

• Sviluppo in serie di Fourier, condizioni di convergenza

$$\text{FORMA ESPONENZIALE: } x(t) = x(t+T) \quad x \in C \quad X(\omega) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} \quad \omega_0 = 2\pi f_0 \quad f_0 = \frac{1}{T} \quad c_n = \frac{1}{T} \int_{-T/2}^{+T/2} x(t) e^{-jnw_0 t} dt$$

Si possono avere più forme di convergenza, rispettivamente quella puntuale e quella in media quadratica:

$$\lim_{N \rightarrow \infty} \left\{ x(t) - \sum_{n=-N}^N c_n e^{jn\omega_0 t} \right\} = 0 \quad \lim_{N \rightarrow \infty} \int_0^T \left| x(t) - \sum_{n=-N}^N c_n e^{jn\omega_0 t} \right|^2 dt = 0$$

In relazione alla prima esistono le *condizioni di convergenza di Dirichelet*: se per il segnale periodico $x(t)$ sono soddisfatte le seguenti condizioni (solo sufficienti)

- 1) $x(t)$ è assolutamente integrabile, cioè $\int_0^T |x(t)| dt < \infty$,
- 2) il numero di massimi e di minimi di $x(t)$ (o meglio della parte reale e della parte immaginaria se $x(t)$ è complesso) in un periodo è finito;
- 3) Il segnale $x(t)$ è continuo o al più presenta un numero finito di discontinuità di prima specie (cioè il limite sinistro e quello destro sono ovviamente diversi nel punto di discontinuità, ma entrambi finiti)

Allora la serie di Fourier converge in modo puntuale a $x(t)$ dove $x(t)$ è continuo, a $\frac{x(t^+) + x(t^-)}{2}$ nei punti di discontinuità di prima specie.

Nel caso in cui l'energia di un segnale su un intervallo di periodicità sia finita, cioè $\int_0^T |x(t)|^2 dt < \infty$ è assicurata (condizione sufficiente) la convergenza in media quadratica.

• Trasformata e antitrasformata di Fourier

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega$$

Analogamente a quanto visto per le serie, la formula di sintesi esprime il segnale $x(t)$ come una somma di infiniti fasori, con pulsazioni (o frequenze) estese ad entrambi i semiassi.

In questo caso però, si tratta di un'infinità non numerabile, e tutte le pulsazioni sono coinvolte, non solo quelle multiple di un determinato valore. Inoltre l'ampiezza di un fasore ad una determinata pulsazione è infinitesima.

Di conseguenza lo spettro di ampiezza bilatero, o meglio la densità spettrale di ampiezza, $|X(\omega)|$ non è più a righe ma continuo. Analogamente è continuo per lo spettro di fase bilatero, definito come $\arg\{X(\omega)\}$

Proprietà della trasformata di Fourier:

- CONIUGAZIONE: $F[x^*(t)] = X^*(-\omega)$

- TRASLAZIONE TEMPORALE: $F[x(t-t_0)] = X(\omega) e^{-j\omega t_0} \rightarrow \text{DIMO斯特RAZIONE: } F[x(t-t_0)] = \int_{-\infty}^{+\infty} x(\xi) e^{-j\omega(\xi+t_0)} d\xi =$

- DERIVAZIONE: esprimiamo $x(t)$ come antitrasformata della sua trasformata

$$F[\dot{x}(t)] = j\omega X(\omega)$$

$$\text{DIMOSTRAZIONE: } \dot{x}(t) = \frac{d}{dt} \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) \frac{d}{dt} e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) j\omega e^{j\omega t} d\omega = F^{-1}[j\omega X(\omega)]$$

$$- INTEGRAZIONE: F\left[\int_{-\infty}^t x(\xi) d\xi\right] = \frac{X(\omega)}{j\omega} \quad \text{se} \quad X(0) = \left[\int_{-\infty}^{+\infty} x(t) dt\right] = 0$$

$$\text{sostituisco } (t-t_0) = \xi$$

$$\int_{-\infty-t_0}^{+\infty-t_0} x(\xi) e^{-j\omega(\xi+t_0)} d\xi =$$

$$= e^{-j\omega t_0} \int_{-\infty}^{+\infty} x(\xi) e^{-j\omega(\xi+t_0)} d\xi = X(\omega) e^{-j\omega t_0}$$

derivo qui l'antitrasformata

- CONVOLUZIONE: date due funzioni $x(t)$ e $y(t)$ aventi come trasformate di Fourier $X(\omega)$ e $Y(\omega)$, il prodotto di convoluzione fra $x(t)$ e $y(t)$ definito come: $x(t)^* y(t) = \int_{-\infty}^{+\infty} x(\tau) y(t-\tau) d\tau$
Ha come trasformata il prodotto delle trasformate.

$$\mathcal{F}[x(t)^* y(t)] = X(\omega) Y(\omega)$$

Dimostrazione: si parte dalla definizione, si esprime quindi $y(t)$ come antitrasformata della sua trasformata, si cambia l'ordine degli integrali ritrovando $X(\omega)$, quindi si nota che il risultato finale è la formula di antitrasformazione di $X(\omega)Y(\omega)$; l'enunciato risulta quindi dimostrato per l'unicità della trasformata (corrispondenza biunivoca).

$$\begin{aligned} x(t)^* y(t) &= \int_{-\infty}^{+\infty} x(\tau) y(t-\tau) d\tau = \int_{\tau} x(\tau) \frac{1}{2\pi} \int_{\omega} Y(\omega) e^{j\omega(t-\tau)} d\omega d\tau = \frac{1}{2\pi} \int_{\omega} Y(\omega) \int_{\tau} x(\tau) e^{-j\omega\tau} d\tau e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{\omega} Y(\omega) X(\omega) e^{j\omega t} d\omega = \\ &= \mathcal{F}^{-1}[Y(\omega)X(\omega)] \quad \downarrow y(t) = \mathcal{F}^{-1}[Y(\omega)] \quad \text{cambio l'ordine degli integrali ritrovando } X(\omega) \end{aligned}$$

• Trasformata della delta di Dirac

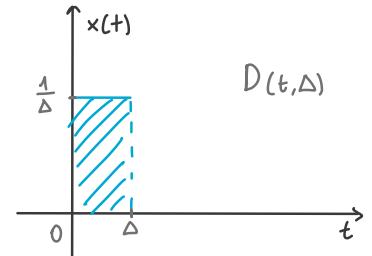
$$\int_{-\infty}^{+\infty} \delta(t) e^{-j\omega t} dt = 1 \quad \text{FUNZIONE GENERALIZZATA DELTA DI DIRAC} \cdot \langle x, \delta \rangle = x(0)$$

Dall'espressione sopra deriva che la trasformata di una delta nell'origine dei tempi corrisponde ad una costante nel dominio delle frequenze.

La dimostrazione risulta immediata dalla definizione di distribuzione, in quanto la funzione esponenziale campionata nell'origine vale 1.

Si noti, come verifica, che lo stesso risultato poteva essere ottenuto utilizzando la funzione ausiliaria $D(t, \Delta)$. Ricordando che

$$\mathcal{F}[D(t, \Delta)] = \frac{\sin \frac{\omega \Delta}{2}}{\frac{\omega \Delta}{2}} e^{-j\frac{\omega \Delta}{2}} \quad \text{SI HA} \quad \lim_{\Delta \rightarrow 0} \int_{-\infty}^{+\infty} D(t, \Delta) e^{-j\omega t} dt = \lim_{\Delta \rightarrow 0} \mathcal{F}[D(t, \Delta)] = 1$$



• Trasformata e antitrasformata di Fourier di una serie, con dimostrazione

Si può definire la trasformata di Fourier della serie temporale mediante la seguente relazione:

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-j\omega nT}$$

La formula di antitrasformazione, che dalla trasformata $X_s(\omega)$ permette di ritornare agli elementi x_n della serie è la seguente:

$$x_n = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{j\omega nT} d\omega \quad n = \dots -2, -1, 0, 1, 2$$

Per la dimostrazione si noti come prima cosa che la trasformata $X_s(\omega)$ è funzione periodica con periodo $\omega_p = \frac{2\pi}{T}$

$$X_s(\omega) = X_s(\omega + \omega_p)$$

Essa può essere espressa mediante uno sviluppo in serie di Fourier di tipo esponenziale. A tale scopo occorre prestare la massima attenzione al significato dei simboli, in quanto la funzione periodica è in ω e non in t .

$$\begin{aligned} X_s(\omega) &= \sum_{n=-\infty}^{\infty} c_n e^{+jn\frac{2\pi}{\omega_p}\omega} = \sum_{n=-\infty}^{\infty} c_n e^{+jn\frac{2\pi}{T}\omega} = \sum_{n=-\infty}^{\infty} c_n e^{+jn\omega T} \\ c_n &= \frac{1}{\omega_p} \int_{-\frac{\omega_p}{2}}^{+\frac{\omega_p}{2}} X_s(\omega) e^{-jn\frac{2\pi}{\omega_p}\omega} d\omega = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{-jn\omega T} d\omega \end{aligned}$$

Confrontando l'espressione sopra dello sviluppo in serie con la definizione di trasformata segue che: $x_n = c_{-n}$

Da cui si ottiene immediatamente l'espressione dell'antitrasformata.

• Risposta impulsiva di una rete causale

Un sistema è detto causale se l'uscita da un determinato istante dipende dai valori passati e da quello attuale dell'ingresso, ma non dai valori futuri; lineare se vale la "sovraposizione degli effetti"; tempo-invariante se la risposta al segnale ritardato è la risposta ritardata, qualunque sia il ritardo t_0 : $y(t - t_0) = \delta[x(t - t_0)]$

La risposta impulsiva $h(t)$ è definita come la risposta della rete all'impulso di Dirac $\delta(t)$ (distribuzione δ). Un sistema lineare tempo invariante può essere completamente caratterizzato dal dominio dei tempi dalla sua $h(t)$. $h(t)$ può essere reale o complessa, noi ci limiteremo al caso reale.

$$\text{CONDIZIONE DI CAUSALITÀ} \rightarrow h(t) = 0 \quad t < 0 \quad h(t) = \lim_{\Delta \rightarrow 0} y_\Delta(t)$$

La risposta impulsiva consente di esprimere l'uscita della rete quando al suo ingresso è presente un generico segnale $x(t)$.

$$y(t) = x(t) * h(t)$$

La dimostrazione di questo fondamentale risultato si basa sul fatto che la Delta di Dirac è l'elemento neutro del prodotto di convoluzione e sul principio di sovrapposizione degli effetti valido per le reti lineari. Per prima cosa si esprime l'ingresso come prodotto di convoluzione di se stesso con la Delta: $x(t) = x(t) * \delta(t) = \int_{-\infty}^{+\infty} x(\tau) \delta(t - \tau) d\tau$

$$\text{per tempo invarianza} \rightarrow \delta(t - \tau) \rightarrow h(t - \tau)$$

$$\text{per linearità (prodotto di una costante)} \rightarrow x(\tau) \delta(t - \tau) \rightarrow x(\tau) h(t - \tau)$$

$$\text{per linearità (somma)} \rightarrow \int_{-\infty}^{+\infty} x(\tau) \delta(t - \tau) d\tau \rightarrow \int_{-\infty}^{+\infty} x(\tau) h(t - \tau) d\tau \quad \text{e per la definizione di convoluzione} \rightarrow x(t) \rightarrow x(t) * h(t)$$

• Risposta di una rete lineare ad una sinusoida

Ad una sinusoida in ingresso una rete lineare tempo-invariante reale risponde con una sinusoida in uscita, avente medesima frequenza angolare e diversa ampiezza e fase.

La risposta dove essere reale (convoluzione di $x(t)$ e $h(t)$ entrambi reali).

$$x(t) = A_x \cos(\omega_n t - \varphi_x) \rightarrow \text{segnale in ingresso}$$

$$A_y = A_x T(\omega_n) \quad \text{ampiezza della sinusoida in uscita}$$

$$\varphi_y = \varphi_x + \beta(\omega_n) \quad \text{fase della sinusoida in uscita}$$

• Energia di un segnale esponenziale monolatero

Possiamo definire l'energia di un segnale come: $E_x \triangleq \lim_{T \rightarrow \infty} \int_{-T/2}^{T/2} |x(t)|^2 dt = \int_{-\infty}^{+\infty} |x(t)|^2 dt$

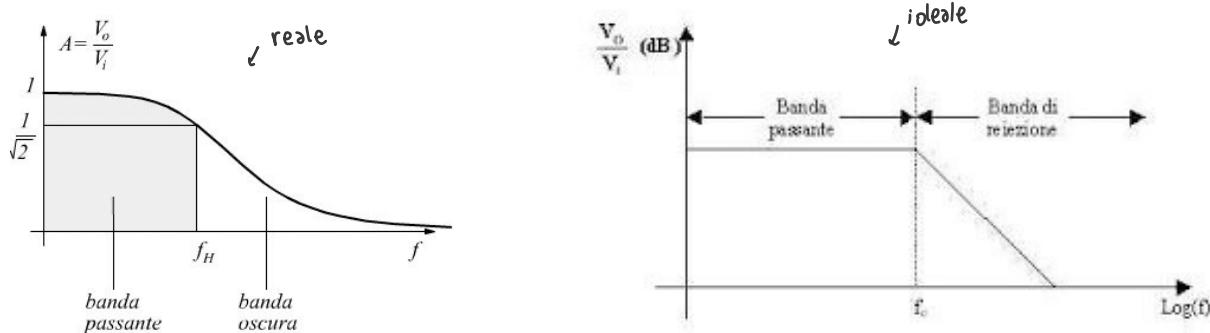
Per l'esponenziale monolatero: $x(t) = A e^{-t/T} u(t)$

$$E_x = \int_0^{+\infty} A^2 e^{-2t/T} dt = \frac{A^2 T}{2}$$

• Se ho un filtro ideale passa-basso con banda 0-4 kHz, e ho in ingresso 10 kHz, il segnale in uscita è distorto?

Un filtro ideale è un sistema lineare avente la proprietà di fare passare alcune componenti in frequenza, senza distorcerle, eliminando le altre. Essi sono caratterizzati da una o più bande passanti, dove valgono le condizioni di non distorsione, e da una o più bande attenuate, dove la caratteristica di ampiezza è nulla. Il passaggio dalle une alle altre avviene alle cosiddette frequenze di taglio. L'idealità nel dominio della frequenza consiste nelle discontinuità della caratteristica di ampiezza nelle transizioni fra bande passanti e bande attenuate (il taglio è verticale, cioè ideale). Essa si riflette nel dominio dei tempi in una risposta impulsiva di tipo anticipativo. Il filtro passa-basso lascia passare indistorte tutte le componenti a frequenza inferiore alla frequenza di taglio, eliminando tutte le altre. È caratterizzato da un'unica frequenza di taglio (sul semiasse positivo).

Mandando in ingresso 10kHz con una frequenza di taglio a 4kHz, il segnale non viene distorto, ma al di sopra della frequenza di taglio il segnale viene fortemente attenuato (equivale a dire che non passa). In un filtro ideale la curva scende in modo lineare (come una retta) dunque lo attenua senza distorcerlo. Nel filtro reale invece la curva scende non proprio come una retta ma quasi. Risulta infatti appena curvilinea e quindi distorce leggermente il segnale, ma rimane sempre una forte attenuazione.

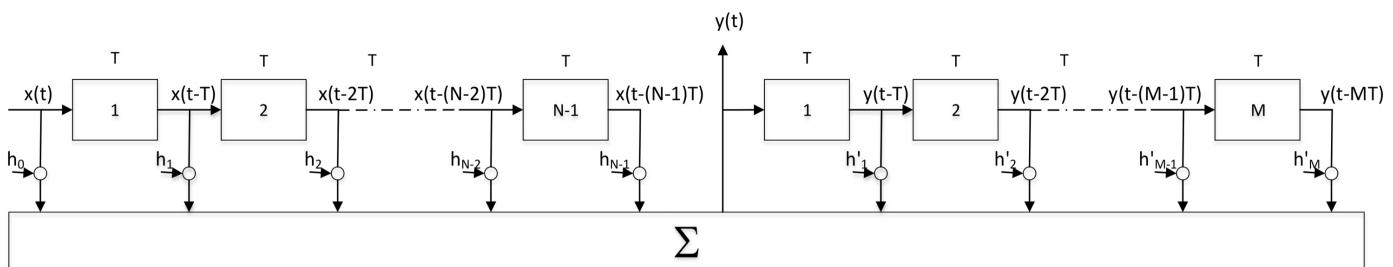


Il filtro passa-basso taglia le frequenze di un segnale oltre la frequenza di taglio, quindi distorce oltre la frequenza di taglio.

• Filtri ricorrenti: cosa succede con il microfono del pc e del telefono entrambi accesi

I filtri ricorrenti seguono un sistema composto da uno schema a blocchi: esso è costituito da due catene di blocchi di ritardo (pari a T), una diretta ed una in retroazione.

La prima è formata da $N-1$ blocchi e da N prese che prelevano versioni diversamente ritardate del segnali d'ingresso; esse moltiplicate per i coefficienti h_k entrano nel sommatore in basso.



Da esso viene prelevata l'uscita $y(t)$ che viene posta in ingresso alla catena di retroazione, simile alla prima ma con M elementi di ritardo ed M prese, da cui vengono prelevate le versioni diversamente ritardate dell'uscita. Una volta moltiplicate per i coefficienti h'_k vengono anche esse inserite nel sommatore.

$$y(t) = \sum_{k=0}^{N-1} h_k x(t-kT) + \sum_{k=1}^M h'_k y(t-kT)$$

Applicando la trasformata di Fourier ad entrambi i membri e ricordando la trasformata di un segnale ritardato si ha:

$$Y(w) = \sum_{k=0}^{N-1} h_k X(w) e^{-j\omega kT} + \sum_{k=1}^M h'_k Y(w) e^{-j\omega kT} = X(w) \sum_{k=0}^{N-1} h_k e^{-j\omega kT} + Y(w) \sum_{k=1}^M h'_k e^{-j\omega kT}$$

Riordinando è immediato ricavare la funzione di trasferimento come rapporto delle trasformate:

$$H(w) = \frac{Y(w)}{X(w)} = \frac{\sum_{k=0}^{N-1} h_k e^{-j\omega kT}}{1 - \sum_{k=1}^M h'_k e^{-j\omega kT}}$$

La funzione di trasferimento dipende dal ritardo T ; dalle lunghezze delle linee dirette e di retroazione, $N-1$, M ; dai valori dei coefficienti. Al variare di tutti questi parametri è possibile fare assumere alla funzione di trasferimento andamenti diversi, mantenendo inalterato lo schema. La complessità del filtro dipende invece dal numero di prese, cioè dalla lunghezza delle linee di ritardo. La funzione di trasferimento è periodica nelle pulsazioni con periodo $\omega_0 = \frac{2\pi}{T}$

Con il microfono del pc e del telefono entrambi accesi, il ritardo dovrebbe produrre interferenza, cioè sovrapposizione dei segnali (nella fattispecie del suono) e quindi fenomeni di risonanza. C'è quindi interferenza dovuta allo sfasamento tra i segnali che quando vanno a sommarsi, sebbene abbiano la stessa frequenza, risultano sfasati.

• Filtro trasversale tempo discreto, funzione di trasferimento e risposta impulsiva

Un filtro trasversale tempo-discreto è costituito da un registro a scorrimento (shift-register) e da un sommatore. Di fatto lo schema è lo stesso del caso tempo continuo, tuttavia in ingresso ed in uscita si hanno in questo caso delle serie temporali. (No retroazione)

Il filtro trasversale è un sistema discreto causale con memoria finita, in quanto non essendoci rami in retroazione, il termine n -esimo della serie temporale d'uscita dipende solamente dal valore in ingresso e dagli $N - 1$ termini precedenti.

La relazione ingresso-uscita è data da:

$$y_n = \sum_{k=0}^{N-1} h'_k x_{n-k}$$

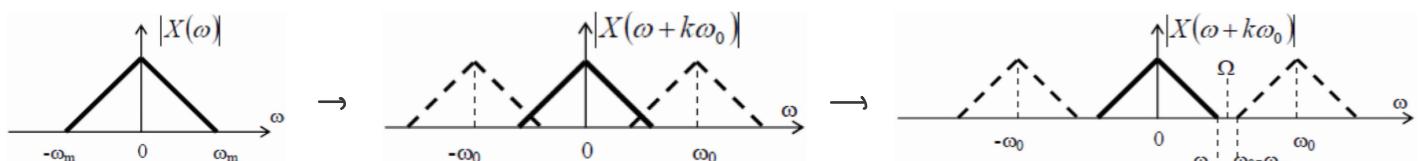
Funzione di trasferimento: $H_s(\omega) = F[\{h_n\}] = \sum_{n=0}^{N-1} h'_n e^{j\omega nT}$ *Risposta impulsiva:* $\{h_n\} = \begin{cases} 0 & n < 0 \\ h'_n & 0 \leq n \leq N-1 \\ 0 & n > N-1 \end{cases}$

• Campionamento tramite Shannon: esprimere il segnale tramite i suoi valori campionati (sviluppo in serie di Shannon), con dimostrazione

L'operazione di campionamento consiste nel leggere i valori di una funzione tempo continua ad intervalli regolari. Converte una funzione tempo continua in una serie temporale. $x_n = x(nT)$

Enunciato del teorema di Shannon: dato un segnale passa-basso, condizione sufficiente perché la conoscenza dei valori campionati sia equivalente alla conoscenza della funzione campionata $x(t)$ è che la frequenza di campionamento sia maggiore del doppio della massima frequenza di $x(t)$. $\omega_0 > 2\omega_m$

Dimostrazione grafica:



Modulo della trasformata del segnale passa-basso

Moduli della trasformata del segnale passa-basso.
Aliasing nelle frequenze.

Moduli della trasformata del segnale passa-basso.
Assenza di aliasing. $\omega_0 > 2\omega_m$

Sotto la condizione sufficiente del teorema di Shannon, la conoscenza dei campioni equivale alla conoscenza della funzione campionata. Lo sviluppo in serie ci permette di esprimere analiticamente $x(t)$ in funzione dei suoi valori campionati $x(nT)$. A tal fine, ricordando che la trasformata della serie equivale alla ripetizione periodica della trasformata del segnale, a meno di una costante moltiplicativa, e scegliendo la frequenza di Nyquist (la metà della frequenza di campionamento), per isolare il termine centrale della ripetizione, abbiamo

$$X(\omega) = \begin{cases} TX_s(\omega) & |\omega| < \frac{\omega_0}{2} \\ 0 & \text{ALTROVE} \end{cases} \rightarrow \text{l'espressione deve essere inserita nella formula di antitrasformazione del segnale } x(t) \rightarrow x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{j\omega t} d\omega$$

Ricordando la definizione di trasformata di Fourier di una serie, sostituendo, e scambiando l'ordine di serie ed integrale, si ottiene:

$$x(t) = \frac{T}{2\pi} \sum_{n=-\infty}^{\infty} x_n \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} e^{j\omega t} e^{-jn\omega T} dw \rightarrow \text{si arriva allo sviluppo in serie di Shannon} \rightarrow x(t) = \sum_{n=-\infty}^{\infty} x_n \frac{\sin \frac{\pi}{T}(t-nT)}{\frac{\pi}{T}(t-nT)} = \sum_{n=-\infty}^{\infty} x_n \operatorname{sinc}\left(\frac{t-nT}{T}\right)$$

Le funzioni che compaiono nello sviluppo in serie sono fra loro ortogonali. I coefficienti dello sviluppo in serie sono quindi i valori campionati.

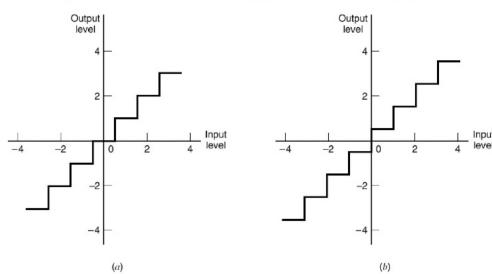
- Se cambio un campione non posso riottenere lo stesso segnale $x(t)$ cambiando gli altri campioni;
- La funzione $\operatorname{sinc}(t/T)$ vale 1 nell'origine e zero per $t=nT$ con $n \neq 0$, segue che per $t=nT$ solo la n -esima funzione è diversa da zero, e vale 1.

• Conversione A/D: quantizzatori

La conversione analogico/digitale PCM (pulse code modulation) prevede tre passaggi: campionamento, quantizzazione e codifica. Il campionamento trasforma il segnale tempo continuo in un segnale tempo-discreto, cioè in una serie temporale. La quantizzazione trasforma la serie dei campioni, continua nei valori, nella serie dei valori quantizzati, discreta nei valori. Ciò consiste in un'approssimazione dei valori ottenuti. L'intervallo di variabilità $[-M, M]$ dei valori campionati viene suddiviso in un numero finito L di intervalli di quantizzazione e tutti i valori interni a ciascuno di questi vengono identificati con uno di essi, indicato con q_n . L'intervallo su cui opera il quantizzatore, $[-M_q, M_q]$ viene detto "dinamica del quantizzatore", ed è opportuno coincida con quella del segnale. Se la dinamica del quantizzatore è suddivisa in intervalli uguali, il quantizzatore si dice uniforme, altrimenti non-uniforme. La differenza fra campione e valore quantizzato corrispondente si dice "errore" di quantizzazione: $e_n = x_n - q_n$

Il legame funzionale fra campioni e valori quantizzati è detto legge di quantizzazione: $q_n = f(x_n)$. I valori assumibili dalla variabile q_n vengono anche detti "livelli" di quantizzazione, per motivi ovvi guardando l'andamento a scala della funzione di quantizzazione. Per la sua semplicità, viene spesso utilizzato il quantizzatore uniforme, anche quando non ottimo. Risulta evidente che maggiore è il numero di livelli L , minore sarà in generale l'errore di quantizzazione. L'ultimo passaggio della conversione, la codifica, trasforma questi valori quantizzati in una serie di bit, associando ad ognuno dei livelli di quantizzazione una parola formata da un certo numero L di bit.

Two types of quantization: (a) midtread and (b) midrise.



• DFT: trasformazione, antitrasformazione, con dimostrazione

La trasformata di Fourier discreta (*DFT = Discrete Fourier Transform*), si applica non più a una serie di infiniti termini ma ad una n-pla, cioè ad un vettore, costituito in generale da componenti complesse. Più precisamente la DFT stabilisce una corrispondenza biunivoca fra n-ple di numeri, in generale complessi: $(x_0, x_1, \dots, x_{N-1}) \xrightarrow{\text{DFT}} (X_0, X_1, \dots, X_{N-1})$

Si noti che i termini sono numerati a partire da 0. L'elemento q-esimo dell'n-pla di arrivo è definito come (formula di trasformazione):

$$X_q = \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi}{N} nq}$$

La formula di antitrasformazione (*IDFT = Inverse Discrete Fourier Transform*), che ci restituisce un termine della n-pla di partenza a partire da quella di arrivo è la seguente:

$$x_n = \frac{1}{N} \sum_{q=0}^{N-1} X_q e^{j \frac{2\pi}{N} nq}$$

Dimostrazione della antitrasformata:

Dal secondo membro della formula di antitrasformazione, sostituendo X_q con la sua definizione (si noti il pedice k anziché n nella sommatoria, per non confonderlo con il pedice di arrivo), cambiando l'ordine delle sommatorie, si ha:

$$\frac{1}{N} \sum_{q=0}^{N-1} X_q e^{j \frac{2\pi}{N} nq} = \frac{1}{N} \sum_{q=0}^{N-1} \sum_{k=0}^{N-1} x_k e^{-j \frac{2\pi}{N} kq} e^{j \frac{2\pi}{N} nq} = \frac{1}{N} \sum_{k=0}^{N-1} x_k \sum_{q=0}^{N-1} e^{j \frac{2\pi}{N} q(n-k)}$$

$\underbrace{X_q}_{X_q}$

Esaminiamo separatamente la seconda sommatoria: $\sum_{q=0}^{N-1} e^{j \frac{2\pi}{N} q(n-k)} = \sum_{q=0}^{N-1} \left(e^{j \frac{2\pi}{N} (n-k)} \right)^q$

Essa può essere vista come la somma parziale di una serie geometrica, di ragione x. Dall'analisi sappiamo che:

$$\sum_{n=0}^{N-1} x^n = \begin{cases} \frac{1-x^N}{1-x} & x \neq 1 \\ N & x = 1 \end{cases} \quad \rightarrow \text{applicando alla nostra serie troncata} \quad \rightarrow \sum_{q=0}^{N-1} \left(e^{j \frac{2\pi}{N} (n-k)} \right)^q = \begin{cases} \frac{1 - e^{j 2\pi (n-k)}}{1 - e^{j \frac{2\pi}{N} (n-k)}} = 0 & n \neq k \\ N & n = k \end{cases}$$

Sostituendo, si ottiene infine:

$$\frac{1}{N} \sum_{q=0}^{N-1} X_q e^{j \frac{2\pi}{N} nq} = \frac{1}{N} N x_n = x_n$$

• Differenza DFT, FFT

Le formule DFT e IDFT richiedono un numero di moltiplicazioni pari ad N^2 . Infatti sono necessarie N moltiplicazioni per ognuno degli N elementi della n-pla di arrivo. Poiché il tempo di calcolo è praticamente determinato da tali moltiplicazioni, possiamo dire che esso cresce proporzionalmente ad N^2 . Tuttavia, nel caso di N potenza di 2, è possibile sfruttare alcune simmetrie nel calcolo ed ottenere lo stesso risultato della DFT con un tempo di calcolo proporzionale a N, conseguendo una fondamentale riduzione.

Gli algoritmi che permettono di conseguire tale risultato sono molteplici: ad essi viene attribuita la denominazione di trasformata di Fourier veloce (FFT = Fast Fourier Transform). Il campo di applicazione della FFT è vastissimo.

• Teorema fondamentale della modulazione

(trasformata del prodotto del segnale con una sinusode)

Dato un segnale $x(t)$ dotato di trasformata $X(\omega)$, calcolare la trasformata $S(\omega)$ della funzione:

$$s(t) = x(t) \cos \omega_0 t$$

Si arriva al risultato esprimendo il coseno come somma di due esponenziali, $\cos \omega_0 t = \frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2}$ ed utilizzando questa formula per scrivere il segnale $s(t)$ come somma di due termini:

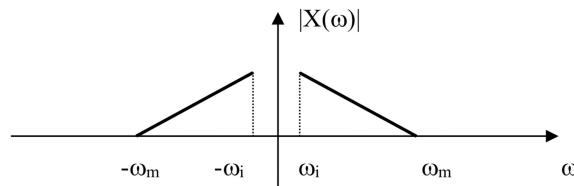
$$s(t) = \frac{1}{2} x(t) e^{j\omega_0 t} + \frac{1}{2} x(t) e^{-j\omega_0 t}$$

Da qui applicando l'operatore di trasformazione ad entrambi i termini (la trasformata di Fourier è lineare, quindi la trasformata di una somma è la somma delle trasformate)

$$\begin{aligned} S(\omega) &= \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{j\omega_0 t} e^{-j\omega t} dt + \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{-j\omega_0 t} e^{-j\omega t} dt = \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{-j(\omega - \omega_0)t} dt + \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{-j(\omega + \omega_0)t} dt = \\ &= \frac{1}{2} X(\omega - \omega_0) + \frac{1}{2} X(\omega + \omega_0) \end{aligned}$$

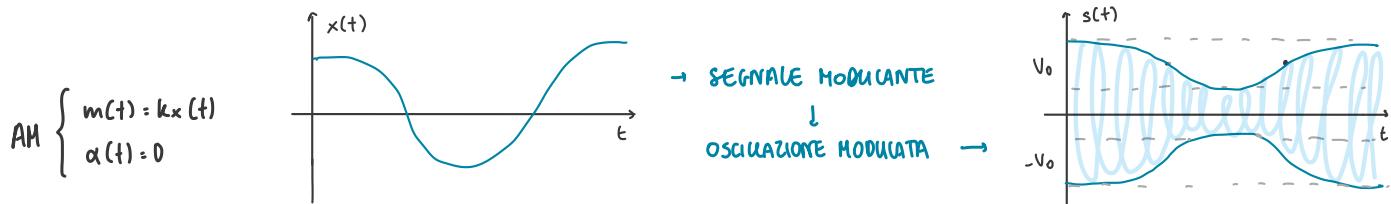
Il risultato finale è fondamentale nella teoria della modulazione ed in particolare nella modulazione lineare (a prodotto e AM). Prende anche il nome in quel contesto di teorema fondamentale della modulazione.

Si noti che la trasformazione di $x(t)$ in $s(t)$ rappresentata dalla moltiplicazione per $\cos \omega_0 t$, nel caso in cui $x(t)$ sia un segnale passa-basso con banda molto minore delle frequenze angolari del coseno (cioè $\omega_0 \gg \omega_m$), come avviene nella modulazione a prodotto (nel qual caso $x(t)$ prende il nome di segnale modulante, $\cos \omega_0 t$ quello di oscillazione portante e $s(t)$ di oscillazione modulata), determina una sostanziale modificazione delle caratteristiche spettrali dei segnali in gioco. Infatti $s(t)$ risulterà una funzione passa-banda, con banda $2\omega_m$ doppia rispetto al segnale passa-basso di partenza, con spettro centrato sulla frequenza ω_0 .



• Modulazione AM, grafico segnale modulante e oscillazione modulata con Ma di 0,5, demodulatore e spettro

La modulazione di ampiezza AM (Amplitude Modulation) è un tipo di modulazione analogica che modifica solo l'ampiezza dell'oscillazione portante.



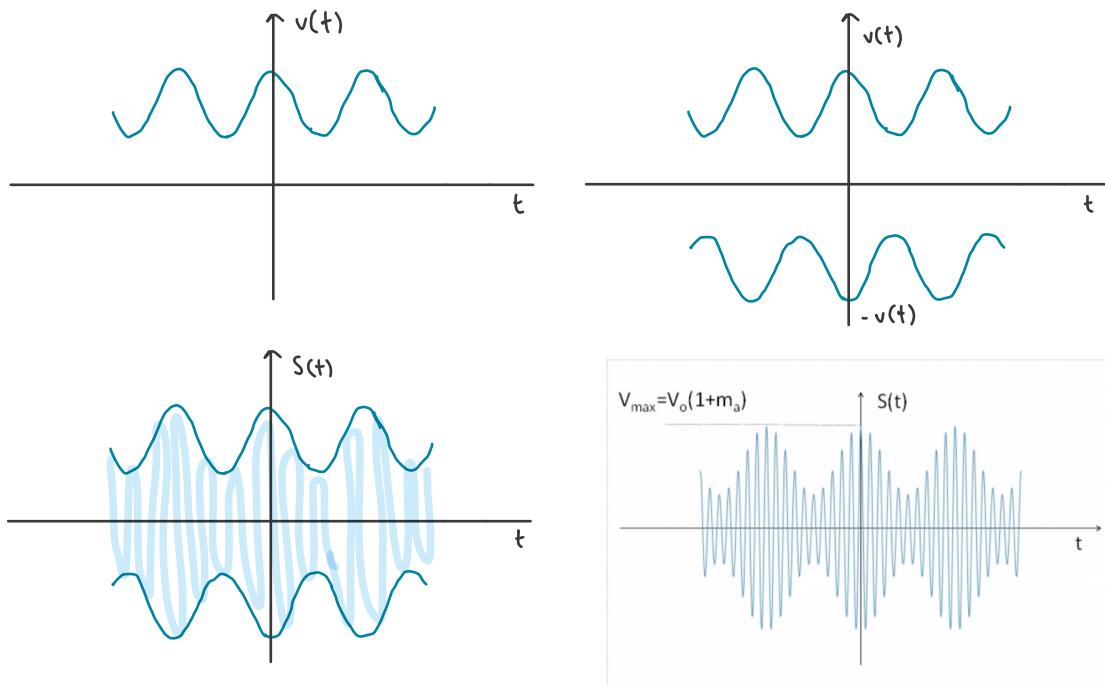
Inoltre in AM $kx(t) \geq -1$ a causa dell'analogo vincolo su $m(t)$, da cui la condizione a destra nella formula: $s(t) = V_0 [1 + kx(t)] \cos [\omega_0 t - \phi_0]$ con $V_0 [1 + kx(t)] \geq 0$ → deve sempre esserci un legame lineare fra ampiezza istantanea e segnale modulante

Indice di modulazione AM: $m_a = \max(|m(t)|) \quad m_a \in [0, 1]$

L'indice di modulazione indica il livello di modulazione (cioè l'entità della trasformazione) che viene impressa alla portante dal segnale modulante. L'indice assume il valore 0 in assenza totale di modulazione, ed il valore 1 quando si ha il massimo della modulazione.

La limitazione a 1 dipende dal fatto che $m(t) \geq -1$ per definizione, essendo $V(t) \geq 0$.

$$\text{con } m_a = 0.5 \quad V_{\max} = V_0 (1 + m_a)$$



segnale modulante sinusoidale

- 1) Disegnare $v(t)$
- 2) Disegnare $-v(t)$
- 3) Disegnare l' oscillazione sinusoidale nel mezzo
↓
si otterrà $S(t)$
(oscillazione modulata)

più aumenta l'indice di modulazione (max 1)
più la portante si alza e si "comprime"

Inviluppo complesso (o equivalente passa-basso) di oscillazioni modulate ("demodulatore"):

Nota la frequenza della portante, l'oscillazione è completamente individuata dal suo inviluppo complesso rappresentativo $i(t)$. Si dice complesso perché anche se l'oscillazione modulata è reale, esso può essere funzione complessa del tempo. Inoltre $i(t)$ è passa-basso anziché passa-banda. Per questo motivo viene anche chiamato "equivalente passa-basso" di $s(t)$. Si noti infine che se non si ha nessuna modulazione, allora l'involuppo complesso da funzione diventa una costante, ricadendo nel caso del metodo simbolico classico.

Consideriamo una nuova espressione generale di un'oscillazione sinusoidale modulata:

$$s(t) = V(t) \cos [\omega_0 t + \alpha(t) - \varphi_0] \longrightarrow s(t) = \operatorname{Re} \{ i(t) e^{j\omega_0 t} \} \quad i(t) = V(t) e^{j[\alpha(t) - \varphi_0]}$$

L'involuppo complesso della somma è dato dalla somma dei due involuppi: $i(t) = i_1(t) + i_2(t)$

Dalle due formule precedenti si vede che il segnale $s(t)$ è anch'esso una oscillazione modulata e che il suo inviluppo complesso è uguale alla somma degli inviluppi complessi delle oscillazioni modulate componenti. Da $i(t)$ è facile ricavare l'ampiezza istantanea e la fase istantanea della modulazione $s(t)$, cioè le "leggi" della modulazione.

$$\text{Si ha infatti: } \begin{cases} V(t) = |i(t)| \\ \alpha(t) = \arg \{ i(t) \} \end{cases}$$

Allo scopo di determinare lo spettro di una modulazione AM consideriamo di nuovo la sua espressione, assunta per semplicità nulla la fase della portante: $s(t) = V_0 [1 + k_x(t)] \cos \omega_0 t \quad V_0 [1 + k_x(t)] \geq 0$

Il suo inviluppo complesso è dato da: $i(t) = V_0 [1 + k_x(t)]$ con $V_0 [1 + k_x(t)] \geq 0$

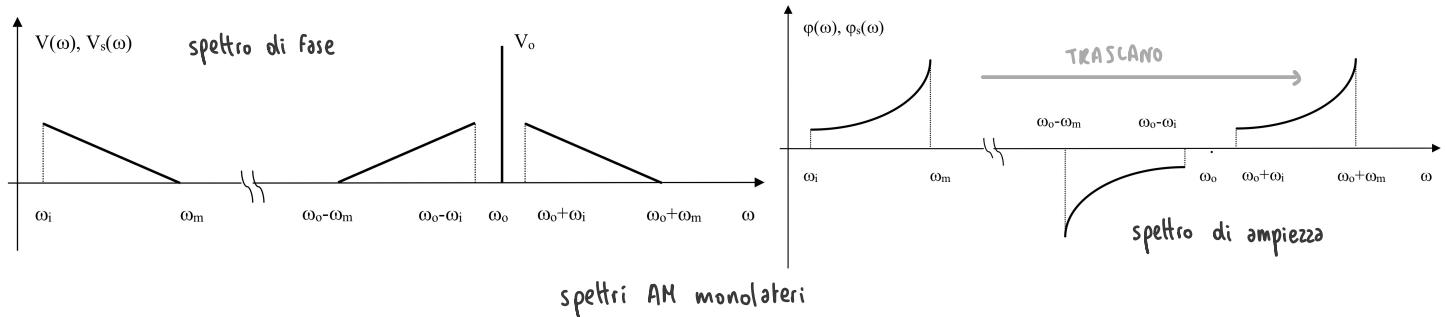
Separando i termini di $s(t)$ possiamo vedere immediatamente che è dato dalla somma della portante e di un termine che, a meno di una costante, è il prodotto fra il segnale modulante e la portante: $s(t) = V_0 \cos \omega_0 t + V_0 k_x(t) \cos \omega_0 t \rightarrow$ gli spettri si possono ottenere anche tramite la trasformata, ma conviene il metodo alt.

Per determinare lo spettro di una modulazione AM consideriamo la rappresentazione monolatera dei segnali. Supponiamo che il segnale modulante $x(t)$, di tipo passa-basso, sia rappresentabile tramite l'integrale di Fourier:

$$s(t) = \int_{\omega_i}^{\omega_m} V(\omega) \cos [\omega t - \varphi(\omega)] d\omega \rightarrow \text{sostituendo nell'espressione dell'AM} \rightarrow s(t) = V_0 \cos \omega_0 t + V_0 k_x(t) \cos \omega_0 t$$

È quindi possibile scomporre l'AM nella somma di tre termini: portante, banda laterale superiore e inferiore.

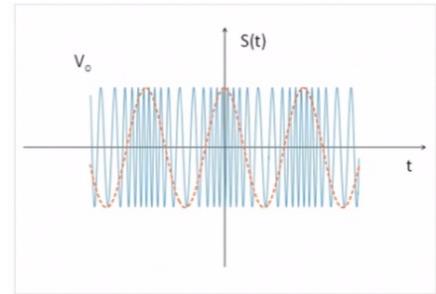
$$s(t) = V_0 \cos \omega_0 t + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega) t - \varphi(\omega)] d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 - \omega) t - \varphi(\omega)] d\omega$$



• Modulazione FM

Viene modificato solo l'argomento dell'oscillazione portante (solo la modulazione in angolo è presente).

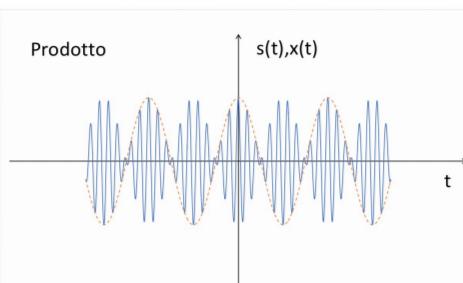
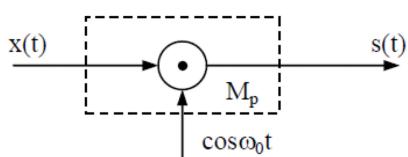
$$\text{FM} \begin{cases} m(t) = 0 \\ \Delta\omega(t) = kx(t) \end{cases} \quad \alpha(t) = k \int_{-\infty}^t x(\tau) d\tau \quad s(t) = V_0 \cos [\omega_0 t + k \int_{-\infty}^t x(\tau) d\tau - \varphi_0]$$



Se $k > 0$, come nel caso considerato, la frequenza istantanea (cioè l'infittimento dell'oscillazione modulata) segue in FM l'andamento del segnale modulante (riportato con linea tratteggiata), quindi è massima in corrispondenza dei massimi del segnale modulante, minima in corrispondenza dei minimi e uguale a quella della portante in corrispondenza degli zeri. In PM segue invece l'andamento della derivata del segnale.

• Modulazione a prodotto, con esempio di modulante sinusoidale, e che succede in caso di errore di fase

La modulazione a prodotto può essere ottenuta direttamente, senza passare dall'AM, facendo il prodotto del segnale modulante $x(t)$ per una sinusode tramite un circuito detto modulatore a prodotto o "mixer", ottenendo: $s(t) = x(t) \cos \omega_0 t$



La modulazione a prodotto è importantissima nella sua versione digitale (ASK, L-ASK).

$x(t)$ tratteggiata (osc. modulante)
 $s(t)$ continua (osc. modulata)

L'inviluppo complesso coincide con il segnale modulante: $i(t) = x(t)$

La modulazione è ibrida. Infatti sia l'ampiezza istantanea che la deviazione istantanea di fase variano nel tempo:

$$V(t) = |x(t)| \quad \alpha(t) = \begin{cases} 0 & x(t) > 0 \\ \pi & x(t) < 0 \end{cases}$$

Gli spettri si ottengono dall'espressione della trasformata: $S(\omega) = \frac{1}{2} X(\omega - \omega_0) + \frac{1}{2} X(\omega + \omega_0)$

La demodulazione si ottiene rimoltiplicando l'oscillazione modulata per la portante (moltiplicata per due per convenienza formale): $u(t) = 2s(t) \cos \omega_0 t = 2x(t) \cos^2 \omega_0 t = x(t) \cos 2\omega_0 t$ da cui filtrando passa-basso: $x_d(t) = x(t)$

Il demodulatore richiede la "ricostruzione" della portante (frequenza e fase comprese), per cui è detto coerente. Il problema è dato dalla ricostruzione della portante, ovvero come è possibile estrarre la portante da $s(t)$. La ricostruzione implica un blocco non lineare in quanto il segnale $s(t)$ non ha una componente alla pulsazione della portante (una rete lineare tempo invariante non può dare in uscita componenti in frequenza che non siano già presenti in ingresso). La portante deve essere ricostruita non solo in frequenza ma anche in fase.

L'errore di fase provoca un'attenuazione del segnale demodulato, che sarà: $x_d(t) = x(t) \cos \Delta$

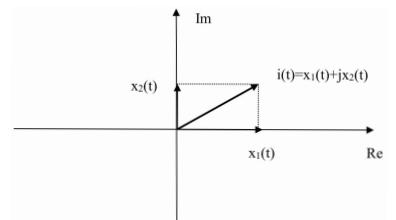
Se infatti la portante ricostruita è sfasata in ritardo si ha: $u(t) = 2s(t) \cos(\omega_0 t - \Delta) = 2x(t) \cos \omega_0 t \cos(\omega_0 t - \Delta) = x(t) \cos \Delta + x(t) \cos(2\omega_0 t - \Delta)$

• Modulazione QAM, inviluppo complesso

Modulatore: è la somma di due modulazioni a prodotto con portanti in quadratura (seconda portante sfasata in anticipo di $\pi/2$). Vi sono due segnali modulanti $x_1(t)$ e $x_2(t)$ aventi le stesse caratteristiche spettrali e indipendenti fra loro: $s(t) = x_1(t) \cos \omega_0 t - x_2(t) \sin \omega_0 t$

L'inviluppo complesso è la somma dei due inviluppi complessi fase è quadratura (indipendenti!): $i(t) = x_1(t) + jx_2(t) = \operatorname{Re} \{(x_1(t) + jx_2(t))(\cos \omega_0 t + j \sin \omega_0 t)\}$

Lo spettro è dato dalla sovrapposizione degli spettri di due modulazioni a prodotto, ma con efficienza in frequenza doppia (1): $n_f = \frac{2Bx}{2Bs} = \frac{2\omega_m}{2\omega_n} = 1$



Demodulatore: per il segnale in uscita al primo (componente in fase) si ha:

$$u_p(t) = 2s(t) \cos \omega_0 t = 2x_1(t) \cos^2 \omega_0 t - 2x_2(t) \sin \omega_0 t \cos \omega_0 t = x_1(t) + x_1(t) \cos 2\omega_0 t - x_2(t) \sin 2\omega_0 t$$

La prima delle tre componenti è l'unica voluta e viene isolata dal filtro passa-basso.

$$x_{pd}(t) = x_1(t) \rightarrow \text{analogamente per la via in quadratura}$$

Nel caso del QAM un **errore di fase** provoca non solo un'attenuazione del segnale utile, ma anche un'interferenza della via in quadratura. Infatti nel caso di una sfasatura in ritardo si ha:

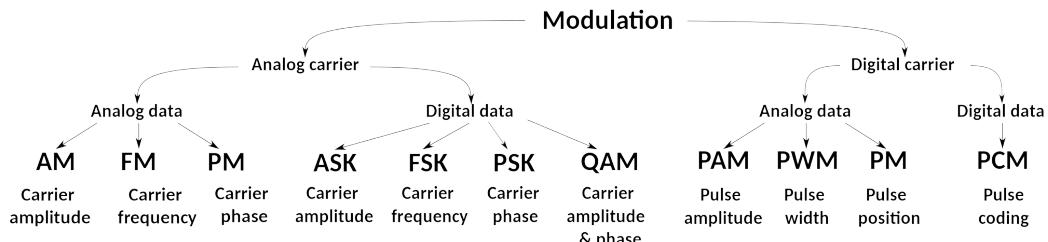
$$x_{pd}(t) = x_1(t) \cos \Delta - x_2(t) \sin \Delta \quad x_{qd}(t) = x_2(t) \cos \Delta$$

• Modulazione QAM, qual è il suo equivalente digitale? $\rightarrow M\text{-QAM} / L\text{-PSK}$

QAM (Quadrature Amplitude Modulation) è uno schema di modulazione utilizzato dagli operatori di rete durante la trasmissione di dati. Si riferisce a un modo per modificare l'ampiezza o il livello di potenza di due segnali. Consente a un segnale analogico di trasmettere in modo efficiente le informazioni digitali e aumenta la larghezza di banda utilizzabile. (molto usato in 802.11 e nell'OFDM)

Le modulazioni digitali si ottengono dalle modulazioni analogiche sostituendo al segnale modulante digitale, cioè di solito un segnale PAM a L livelli con codifica NRZ.

Analogica	Digitale
AM, o DSB	
Prodotto o DSB-SC	ASK, L-ASK
QAM	M-QAM, QPSK
PM	PSK, BPSK, 2-PSK, L-PSK
FM	L-CP-FSK, (MSK)
	FSK, L-FSK



• Funzioni di crosscorrelazione e autocorrelazione

Segnali ad energia finita:

Dati due segnali in generale complessi, $x(t)$ ed $y(t)$, si definisce funzione di crosscorrelazione il coniugato del prodotto interno di un uno di essi per la versione anticipata dell'altro:

$$\varphi_{xy}(\tau) \triangleq x, y_\tau >^* = \int_{-\infty}^{+\infty} x^*(t) y(t+\tau) dt$$

Nel caso particolare in cui $y(t)=x(t)$, la funzione di crosscorrelazione prende il nome di funzione di autocorrelazione, che risulta quindi così definita:

$$\varphi_x(\tau) \triangleq x, x_\tau >^* = \int_{-\infty}^{+\infty} x^*(t) x(t+\tau) dt$$

Si noti che l' autocorrelazione calcolata nell'origine rappresenta l'energia di un segnale.

$$\varphi_x(0) = \int_{-\infty}^{+\infty} x^*(t) x(t) dt = \int_{-\infty}^{+\infty} |x(t)|^2 dt = E_x$$

Segnali a potenza finita:

Le definizioni delle funzioni di crosscorrelazione ed autocorrelazione per le funzioni a potenza finita si possono ottenere con la seguente sostituzione formale:

$$\int_{-\infty}^{+\infty} dt \rightarrow \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} dt$$

crosscorrelazione: $\varphi_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) y(t+\tau) dt$
autocorrelazione: $\varphi_x(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) x(t+\tau) dt$

Per le funzioni appena definite valgono le seguenti proprietà, analoghe a quelle dei segnali ad energia finita.

- $\varphi_{xy}(\tau) = \varphi_{y,x}^*(-\tau)$
- $\varphi_x(\tau) = \varphi_x^*(-\tau)$
- $\varphi_x(\tau) = \varphi_x(-\tau)$ se $x(t)$ è reale
- $|\varphi_{xy}(\tau)|^2 \leq \varphi_x(0)\varphi_y(0) = P_x P_y$
- $|\varphi_x(t)| \leq \varphi_x(0) = P_x$

L'ultima relazione in particolare afferma che il massimo (in modulo) della funzione di autocorrelazione è nell'origine, e che esso coincide con la potenza.

• Funzione di autocorrelazione nei segnali a potenza finita e periodici

I segnali periodici rappresentano un caso particolare di segnali a potenza finita, in quanto per essi è possibile avere altre rappresentazioni nel dominio delle frequenze (sviluppi in serie di Fourier, trasformata generalizzata). Risulta interessante individuare comunque anche il loro spettro di potenza, per collegarlo alle altre rappresentazioni. Si noti innanzitutto che le funzioni di cross ed autocorrelazione di segnali periodici di ugual periodo T , possono essere calcolate su un periodo. Applicando quindi la seguente sostituzione formale:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} dt \rightarrow \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} dt$$

crosscorrelazione: $\varphi_{xy}(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) y(t+\tau) dt$
autocorrelazione: $\varphi_x(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) x(t+\tau) dt$

• Capitolo 9: autocorrelazione, segnali PAM

I segnali PAM (Pulse Amplitude Modulation) sono esprimibili come convoluzione fra una serie temporale $\{a_n\}$ ed un impulso ad energia finita $g(t)$:

$$s(t) = \{a_n\} * g(t) = \sum_{n=-\infty}^{+\infty} a_n g(t-nT)$$

impulso rettangolare, di ampiezza unitaria, con origine a $t=0$

Il segnale PAM può essere analogico, se la serie è analogica, ovvero se ogni termine della serie può assumere con continuità infiniti valori all'interno di un prefissato intervallo, digitale se invece può assumere solo un numero finito di valori.

Se la serie temporale è ad energia finita, allora anche il segnale PAM è ad energia finita, e la sua trasformata è data dal prodotto delle trasformate: $S(\omega) = A_s(\omega) G(\omega)$

Se invece la serie temporale è a potenza finita, il segnale PAM è anch'esso a potenza finita e non è trasformabile secondo Fourier, per cui occorre passare ad un'analisi generalizzata, ovvero calcolare il suo spettro di potenza.

Consideriamo qui il caso dei segnali PAM a potenza finita per analizzarne lo spettro. Assumiamo la serie a valore medio nullo per semplicità; si può dimostrare che la sua funzione di autocorrelazione è data dal prodotto di convoluzione delle funzioni di autocorrelazione della serie e dell'impulso, a meno di una costante $1/T$.

$$\Psi_s(\tau) = \frac{1}{T} \{\psi_u\} * \langle \psi_g(\tau) \rangle$$

$$C_k = \langle x_n * x_{n+k} \rangle = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} x_n * x_{n+k}$$

Lo spettro di potenza del segnale PAM è dato dalla trasformata della funzione di autocorrelazione (nell'ipotesi che esista):

$$G_{s,bil}(\omega) = \frac{F[\Psi_s(\tau)]}{2\pi} = \frac{1}{2\pi T} F[\psi_g(\tau)] F[\{\psi_u\}] = \frac{|G(\omega)|^2}{2\pi T} \sum_{n=-\infty}^{\infty} c_n e^{-jk\omega T}$$

Si noti che lo spettro è dato dal prodotto di due termini: il primo dipende dalla forma dell'impulso, il secondo dalla autocorrelazione della serie.

Se la serie temporale è reale, la sua autocorrelazione è reale e pari, per cui la sua trasformata può essere scritta con solo riferimento agli indici positivi:

$$\sum_{k=-\infty}^{\infty} c_k e^{-jk\omega T} = c_0 + 2 \sum_{k=1}^{\infty} c_k \cos k\omega T \rightarrow G_{s,bil}(\omega) = \frac{|G(\omega)|^2}{2\pi T} \left[c_0 + 2 \sum_{k=1}^{\infty} c_k \cos k\omega T \right]$$

Infine se la autocorrelazione è nulla per k diverso da zero, come spesso in pratica, lo spettro di potenza del segnale PAM coincide con lo spettro di energia dell'impulso, a meno di una costante moltiplicativa. $G_{s,bil}(\omega) = \frac{c_0 |G(\omega)|^2}{2\pi T}$

Un segnale è aleatorio quando il suo andamento nel tempo è aleatorio, ovvero è una funzione aleatoria (detta anche processo stocastico). Un segnale PAM è aleatorio se lo è la serie temporale $\{a_n\}$. Per una serie aleatoria non è nota a priori la successione dei valori; tuttavia ciò non significa che la conoscenza della serie sia nulla, infatti la serie può sempre essere descritta in termini statistici, come successione di infinite variabili aleatorie; in particolare se la serie è stazionaria tutte le valutazioni statistiche sono indipendenti dall'indice n che denota la posizione della variabile all'interno della sequenza.

Ad esempio, se la serie temporale fosse data dai lanci successivi di un dado, il risultato dell' n -esimo lancio sarebbe dato dalla variabile aleatoria a_n . È utile la formula del valore medio statistico di una serie stazionaria: $E[a_n] = \sum_{i=1}^L a_i P(a_i)$

(gli a_i rappresentano i valori che possono essere assunti e le $P(a_i)$ le rispettive probabilità) Poiché queste ultime non dipendono da n , per la stazionarietà, anche il valore medio è indipendente da n .

Per le serie aleatorie è possibile definire una autocorrelazione statistica (a priori) come media statistica del prodotto delle coppie di valori posti a distanza k, ovvero come $E[a_n * a_{n+k}]$; se la serie è stazionaria, come supporremo d'ora in avanti, la probabilità della coppia (congiunta del secondo ordine) non dipende dalla posizione delle due variabili aleatorie all'interno della sequenza, cioè dal pedice "n" che rappresenta la posizione del primo elemento, ma solo dalla distanza fra gli elementi, cioè da "k":

$$C_{\text{stat},k} = E[a_n * a_{n+k}] = \sum_{i=1}^L \sum_{\ell=1}^L (a_i)^* a_\ell^k P(a_i, a_\ell, k)$$

potenza statistica
↑

Per $k = 0$ coincide con il momento del secondo ordine: $C_{\text{stat},0} = E[a_n * a_n] = \sum_{i=1}^L (a_i)^* a_i^0 P(a_i) = \sum_{i=1}^L |a_i|^2 P(a_i)$

Condizione sufficiente per l'incorrelazione è che le variabili aleatorie a_n e a_{n+k} siano indipendenti. Se inoltre il valor medio è nullo, cioè $E[a_n] = 0$, le variabili sono incorrelate se e solo se l'autocorrelazione è nulla per k diverso da zero.

Infine se la serie è "ergodica" (è sufficiente che oltre che stazionaria sia a memoria finita, cioè la variabili a_n e a_{n+k} siano indipendenti per valori di k sufficientemente grandi) i valori medi statistici coincidono con gli analoghi temporali che possono essere calcolati a posteriori sulle realizzazioni del processo stocastico, cioè sulle serie temporali che si ottengono a posteriori, come risultato dell'esperimento (ad esempio nel caso del dado, una realizzazione è l'effettiva sequenza dei valori ottenuta lanciando il dado consecutivamente). Si ha quindi in particolare che la autocorrelazione statistica e quella temporale vengono a coincidere per i processi ergodici: $C_{\text{stat},k} = c_k$

- Tutte le realizzazioni di un processo ergodico, anche se ovviamente diverse fra loro, hanno la stessa autocorrelazione temporale, dovendo questa coincidere con quella statistica, che è unica;
- Ciò implica che tutte le realizzazioni dei segnali PAM, pur diverse fra loro in quanto è diversa la successione dei simboli della serie, hanno lo stesso spettro di potenza; esso si ottiene dalla formula per i segnali deterministicamente sostituendo alla autocorrelazione temporale quella statistica, data dalla definizione di autocorrelazione statistica: $c_k = E[a_n * a_{n+k}] = \sum_{i=1}^L \sum_{\ell=1}^L (a_i)^* a_\ell^k P(a_i, a_\ell, k)$

Se la serie è a valor medio nullo e gli elementi della serie aleatoria sono incorrelati (come detto condizione sufficiente è che siano indipendenti) si ottiene: $G_{s,\text{bil}}(\omega) = \frac{E[|a_n|^2]}{2\pi T} |G(\omega)|^2$

• PAM dove viene utilizzato?

La modulazione PAM (*Pulse Amplitude Modulation*) ovvero modulazione di ampiezza di impulso codifica l'informazione nell'ampiezza di una serie di segnali. La modulazione PAM viene utilizzata molto nelle comunicazioni via cavo, ad esempio le trasmissioni televisive terrestri e i vari standard Ethernet. In particolare, lo standard 100BASE-T4 utilizza una modulazione PAM a 3 livelli, 100BASE-T ovvero Gigabit Ethernet utilizza PAM a 5 livelli e 10 Gigabit Ethernet usa PAM-16.

25 Gigabit Ethernet e alcune varianti di 100 Gigabit Ethernet e 200 Gigabit Ethernet usano una modulazione PAM-4.

La modulazione PAM viene utilizzata anche in alcune schede grafiche Nvidia, nella foto biologia, nei driver elettronici utilizzati per l'illuminazione LED.

• Codifica multilivello

La codifica multilivello è un'estensione della codifica bipolare, nella quale viene emesso un simbolo ogni l bit, per cui si ha: $T = lT_b$ $f_s = \frac{1}{T} = \frac{f_b}{l}$

I simboli possono assumere i seguenti $L = 2^l$ valori: $a^l = \pm 1, \pm 3, \dots, \pm (L-1)$

- Il caso bipolare è un caso particolare di codifica multilivello ($L=2$).
- Numeri dispari per avere la stessa distanza (2) dal valore successivo.
- Non è obbligatorio, ma conviene mappare gruppi di l bit in un simbolo adottando una codifica di Gray; a simboli contigui corrispondono l-piè che differiscono di un solo bit.

<i>l-pla</i>	Simbolo
000	+7
001	+5
011	+3
010	+1
110	-1
111	-3
101	-5
100	-7

Nella codifica multilivello i simboli a_n sono indipendenti ed equiprobabili, ottenuti da gruppi di l bit distinti (senza bit in comune). Il valore medio è quindi nullo mentre la funzione di autocorrelazione statistica è data da:

$$C_{\text{stat}, k} = \begin{cases} E[a_n^k] = \frac{L^k - 1}{3} & k = 0 \\ E[a_n]E[a_n] = 0 & k \neq 0 \end{cases}$$

Nel caso di codifica multilivello (comprendente anche il caso bipolare), la autocorrelazione è nulla tranne che nell'origine. Lo spettro del segnale PAM diventa quindi:

$$G_{s,\text{bil}} = \frac{|G(\omega)|^2}{2\pi T} \quad E[a_n^2] = \frac{L^2 - 1}{3}$$

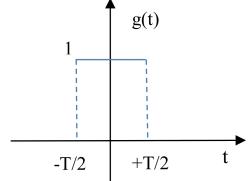
Come impulso $g(t)$ si può prendere l'impulso rettangolare di durata T (detto anche NRZ, No Return to Zero);

$$\text{La sua trasformata è data da: } G(\omega) = T \frac{\sin \omega T / 2}{\omega T / 2} \rightarrow G_{s,\text{bil}}(\omega) = \frac{E[a_n^2]T}{2\pi} \left| \frac{\sin \omega T / 2}{\omega T / 2} \right|^2$$

$$\text{Conviene passare alle frequenze: } G_{s,f,\text{bil}}(f) = 2\pi G_{s,\text{bil}}(2\pi f) = E[a_n^2]T \left| \frac{\sin 2\pi f T / 2}{2\pi f T / 2} \right|^2 = E[a_n^2]T |\text{sinc} f T|^2 = \frac{E[a_n^2]}{f_s} \left| \text{sinc} \frac{f}{f_s} \right|^2$$

- La frequenza di simbolo coincide con la frequenza di bit nel caso bipolare ($l=1$), altrimenti è ridotta di un fattore l.
- La banda del primo lobo è pari alla frequenza di simbolo.

Vantaggio del multilivello: a parità di frequenza di bit la banda è ridotta di un fattore l;
Svantaggio: più alta probabilità d'errore a parità di altre condizioni.



• OFDM nel Wi-Fi, varie versioni del Wi-Fi

L'OFDM (Orthogonal Frequency-division multiplexing) è una tecnica di multiplazione di frequenza per segnali digitali, basata sull'utilizzo combinato di un insieme di sottoportanti fra loro ortogonali. Lo standard iniziale (1997) di 802.11 (wifi) definì una LAN wireless a 1 o 2 Mbps che saltava tra le frequenze per diffondere il segnale, lo velocizzarono a 11 Mbps con 802.11b (1999). L'OFDM fu introdotto con 802.11a (1999) e 802.11g (2003), in cui la modulazione funzionava dividendo una banda larga spettrale in N sotto portanti e qui distanziate $\Delta f = B/N$. N è la dimensione della FFT. Ogni portante sarà modulata a L livelli quindi porterà m bit per ogni tempo di simbolo OFDM. Nel Wifi m è lo stesso per tutte le portanti, mentre nell'ADSL e in altri casi no. Grazie a questo meccanismo, si raggiunse una velocità di 54 Mbps. Nel 2009 IEEE ratificò 802.11 n, che utilizzava quattro antenne per trasmettere 4 flussi di dati contemporaneamente, con una velocità di 100 Mbps. Nel 2021 → 802.11ax → OFDMA ovvero garantisce la contemporaneità, non con le antenne ma con le frequenze pilota.

• Problema dell'accesso multiplo, algoritmi di collisione (CSMA/CD e CSMA/CA)

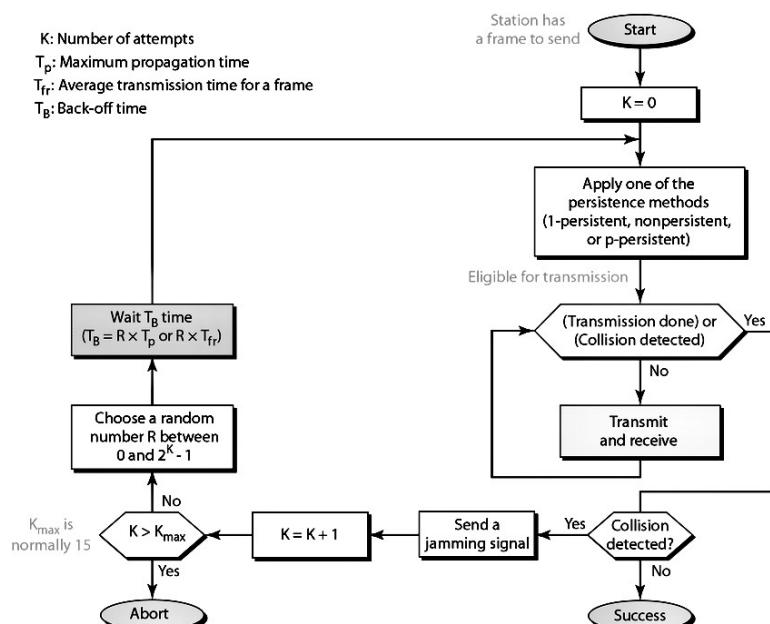
Quando l'accesso ad una risorsa può avvenire da parte di più utenti indipendenti, si parla di risorsa condivisa ed è necessaria l'implementazione di particolari protocolli di accesso multiplo. Nelle reti locali tutte le stazioni condividono lo stesso canale trasmissivo, generalmente ad alta velocità. Quando una stazione ottiene l'accesso alla trasmissione, essa occupa temporaneamente tutta la banda disponibile per il tempo necessario a trasmettere uno o più pacchetti. I pacchetti immessi sulla rete sono ricevuti da tutte le stazioni presenti sulla LAN e perciò la trasmissione è di tipo "broadcast". Per assicurarsi che un dispositivo non cerchi di utilizzare la rete se un altro elaboratore la sta già utilizzando, si utilizza il protocollo CSMA (acronimo di Carrier Sense Multiple Access, ovvero accesso multiplo tramite rilevamento della portante).

Nelle reti LAN cablate, ad esempio Ethernet, viene utilizzata la versione **CSMA/CD 1-persistente** (Collision detection, rilevamento della collisione): il protocollo implementa la direttiva "ascolta prima di trasmettere e mentre trasmetti; se mentre trasmetti rilevi collisioni, fermati, segnala a tutte le altre stazioni la collisione e riprova più tardi secondo modalità di ritrasmissione stabilita". L'algoritmo è il seguente: l'adattatore sistema il frame da trasmettere in un buffer; se il canale è inattivo si attende un tempo di 96 bit-time e si procede alla trasmissione, se invece è occupato si attende che il canale torni libero prima di ritrasmettere; durante la trasmissione l'adattatore monitora la rete: se non riceve segnali da altri adattatori considera il frame spedito; se l'adattatore riceve un segnale da un altro adattatore, arresta la trasmissione e trasmette un segnale di disturbo (*jamming signal*) che avverte le altre stazioni dell'avvenuta collisione bloccandone la trasmissione.

Condizione necessaria per la rilevazione della collisione da parte di una stazione trasmittente è che il tempo di trasmissione T_t sia maggiore o al più uguale al tempo massimo di andata e ritorno (RTT Round trip Time): $T_t = \frac{L}{C} \geq RTT = \frac{2D}{V}$

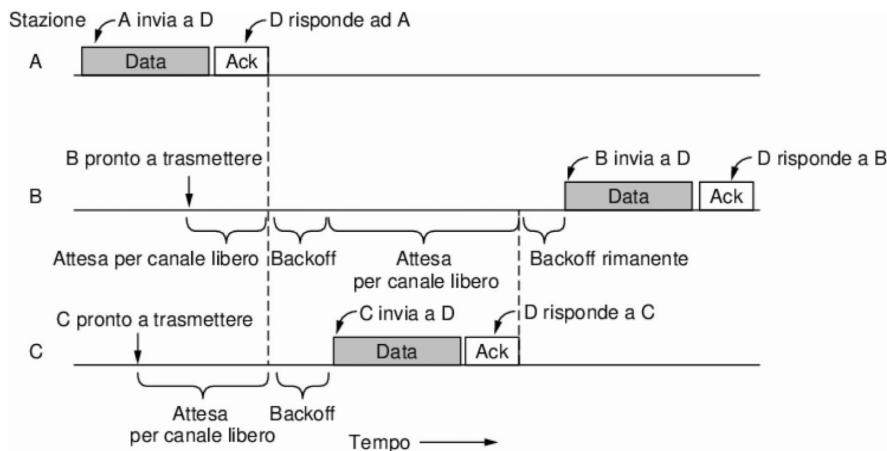
dove L è la lunghezza della trama di dimensione minima, C è la velocità di trasmissione, D la lunghezza del segmento di rete e V la velocità di propagazione del segnale nel mezzo.

Il CSMA/CD era utilizzato nelle ormai obsolete varianti di Ethernet su mezzo condiviso (10BASE5, 10BASE2) e in quelle su rame che usavano gli hub. Le moderne reti Ethernet, che fanno uso di switch con connessioni full-duplex, non usano il CSMA/CD in quanto i segmenti sono isolati.



Nelle reti LAN wireless, ad esempio 802.11 (WiFi) si utilizza il protocollo di accesso multiplo **CSMA/CA** (Collision Avoidance, evita le collisioni) che utilizza il rilevamento della portante ma in cui i nodi tentano di evitare a priori il verificarsi di collisioni. Una volta iniziata, la trasmissione prosegue fino al termine del pacchetto. È di particolare importanza nei casi in cui il rilevamento delle collisioni non è realizzabile, come avviene in pratica nelle reti senza fili. Infatti, oltre all'esistenza del problema dei nodi nascosti, il ricevitore di un nodo non può rilevare in modo affidabile eventuali trasmissioni provenienti da altri nodi mentre il relativo trasmettitore è attivo.

In 802.11, nel momento in cui una stazione vorrebbe tentare una trasmissione, essa ascolta il canale (listen-before-transmit). Se il canale risulta libero, la stazione attende per un certo lasso di tempo identificato come DIFS (Distributed Inter Frame Space) trascorso il quale, se il canale continua ad essere libero, la stazione inizia la trasmissione del pacchetto. A trasmissione completata il nodo di trasmissione attende per un tempo detto SIFS (short Inter frame space, di durata inferiore al DIFS) la ricezione di un ACK che conferma dell'avvenuta ricezione da parte della stazione ricevente. Durante la trasmissione dello SIFS le altre stazioni, trovando il canale occupato, non trasmetteranno, evitando in tal modo collisioni.



Qualora invece la stazione trasmittente rilevi il canale occupato oppure si siano verificate delle prenotazioni da parte di altre stazioni, la stazione attenda per una durata casuale (tempo di backoff) che il canale si liberi. Questa attesa è implementata per mezzo di un timer che viene decrementato solo durante i periodi di inattività del canale, mentre viene invece congelato durante i restanti periodi di trasmissione sul canale da parte di altre stazioni.

Spesso il protocollo CSMA/CA è usato congiuntamente alla tecnica RTS/CTS (prenotazione del canale) per affrontare il problema del cosiddetto terminale nascosto.

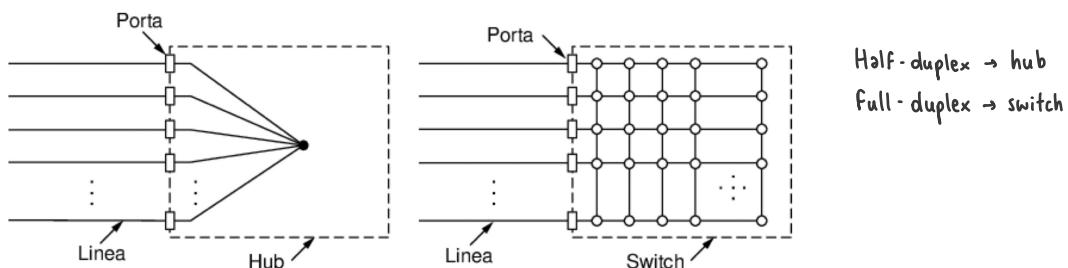
• Quando in Ethernet ho il problema di accesso condiviso e quando no?

Nelle prime versioni ormai obsolete di Ethernet, con cavo coassiale o provviste di hub tutte le stazioni sono nello stesso dominio di collisione, quindi è necessario utilizzare l'algoritmo di collisione CSMA/CD per pianificare le trasmissioni.

Nelle moderne versioni di switched Ethernet, provviste quindi di switch (commutatore), ogni porta ha il proprio dominio di collisione indipendente; nel caso comune in cui il cavo sia full duplex, sia la stazione sia la porta possono spedire il frame lungo il cavo contemporaneamente, senza preoccuparsi delle altre stazioni e porte, rendendo inutile l'utilizzo di CSMA/CD in quanto le collisioni sono ora impossibili; tuttavia, se il cavo è half duplex, la stazione e la porta si contendono la porta come al solito con CSMA/CD.

• Differenza fra switch e hub

Un hub connette elettricamente tutti i cavi collegati come se fossero saldati assieme, e tutte le stazioni sono nello stesso dominio di collisione (CSMA/CD). Lo switch invece contiene una scheda hardware di collegamento tra le interfacce di rete o backplane ad alta velocità che connette tutte le porte. Dall'esterno uno switch assomiglia a un hub. Sono entrambi scatole, tipicamente dotati di un numero di porte da 4 a 48, ognuna fornita di un connettore standard RJ-45 per doppino. Ogni cavo connette lo switch o un hub a un solo computer. Dentro lo switch, tuttavia, accade qualcosa di molto diverso. Gli switch devono solo far uscire i frame verso le giuste porte di destinazione. Quando una porta dello switch riceve un frame Ethernet da una stazione, lo switch verifica l'indirizzo Ethernet per capire a quale porta sia destinato. Ogni porta ha il proprio dominio di collisione indipendente.



Un hub, o *ripetitore*, che si trova al livello 1 non tratta frame o pacchetti, ma solo bit. Inoltre ogni bit a tutte le porte, e non ha capacità di filtraggio.

Gli switch Ethernet sono un modo attuale di chiamare i bridge; forniscono funzionalità che vanno oltre quelle di Ethernet classica e gli hub Ethernet per facilitare l'unione di più LAN in una rete più grande. (bridge = switch)

I bridge operano a livello 2, quindi esaminano gli indirizzi del livello data link per inoltrare i frame (non i pacchetti). Con un bridge si possono unire due LAN separate, raggiungendo il doppio della capacità di una singola LAN pur mantenendo questa capacità. Il bridge possiede una tabella di switch e usa gli indirizzi MAC per trovare la porta a cui inoltrare il frame di input, con capacità di filtraggio.

Nel *learning bridge* si parte da una tabella vuota, che si riempie automaticamente man mano che “conosce” gli indirizzi delle varie porte a cui invia/riceve frame.

• Differenza tra CSMA/CD e CSMA/CA

Il CSMA/CD viene utilizzato nelle LAN cablate, invece il CSMA/CA utilizzato nelle LAN wireless e altri tipi di reti wireless. Inoltre, CSMA/CD è standardizzato in IEEE 802.3 mentre CSMA/CA è standardizzato in IEEE 802.11. Concettualmente sono due protocolli simili, con il rilevamento della portante prima dell'invio e il backoff esponenziale dopo le collisioni; tuttavia una stazione con un frame da spedire in CSMA/CD inizia con un backoff casuale e non aspetta che avvenga una collisione, poiché la natura del canale wireless preclude la possibilità di utilizzare meccanismi affidabili di rilevamento di collisioni. Far partire presto i backoff aiuta a evitare le collisioni, e poiché non possono essere rilegati, gli acknowledgement sono usati per dedurre le collisioni.

Sostanzialmente, CSMA/CD agisce dopo aver rilevato una collisione, riduce soltanto il tempo di recupero e ritrasmette i frame appena trova un conflitto; CSMA/CD agisce per evitare le collisioni, per minimizzarne le possibilità, trasmettendo in primis “l'intento” di inviare una trasmissione di dati.

Questo modo di operare è chiamato DCF (*Distributed coordination function*) perché ogni stazione agisce indipendentemente, senza alcun controllo centrale.

• 802.11 alternative all' accesso a contesa

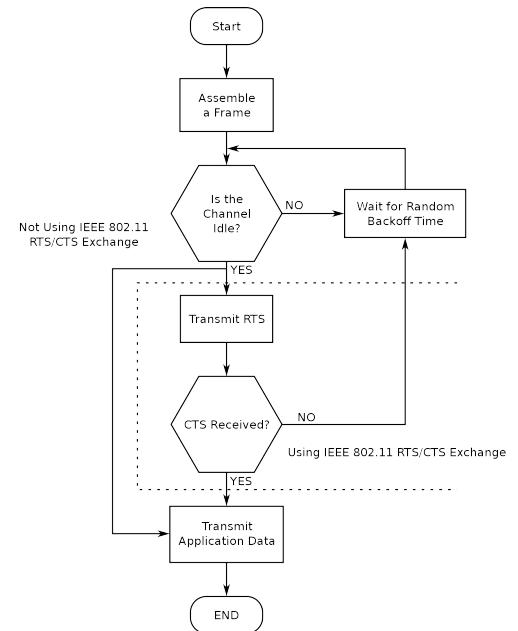
Oltre al DCF, lo standard include anche un modo di operare opzionale chiamato PCF (*Point Coordination Function*) che definisce regole di accesso al canale differenti, basate sul *polling* delle stazioni da parte di un *Point Coordinator (PC)* che risiede sull'access point. In questo caso è l'AP che mantiene il controllo della rete e decide periodicamente a quale stazione assegnare il canale in base alla sua polling list. PCF è però un meccanismo opzionale e poco implementato nella realtà.

• Problema stazione nascosta

Un problema di natura wireless è che i raggi di trasmissione di stazioni diverse possono essere differenti, cosa che non accade con le reti via cavo. Di conseguenza, possono sorgere situazioni come il problema del terminale nascosto: poiché non tutte le stazioni sono a reciproca portata radio, le trasmissioni che avvengono in una parte di una cella potrebbero non essere ricevute in qualche altro punto nella stessa cella. Ad esempio, se la stazione C sta trasmettendo alla stazione B, A rileva la portante sul canale, non sentirà nulla e dedurrà di poter trasmettere a B, generando una collisione.

Per far fronte a questo problema, IEEE 802.11 definisce la rilevazione del canale come composta sia dalla rilevazione fisica che da quella virtuale. La rilevazione fisica controlla il mezzo trasmissivo per capire se c'è un segnale valido. Con la rilevazione virtuale ogni stazione registra quando il canale è occupato tenendo traccia dei NAV (*network allocation vector*): ogni frame possiede un campo NAV che indica quanto tempo sia necessario per completare la sequenza, così le stazioni che vedono il frame sapranno che il canale resterà occupato per quel lasso di tempo.

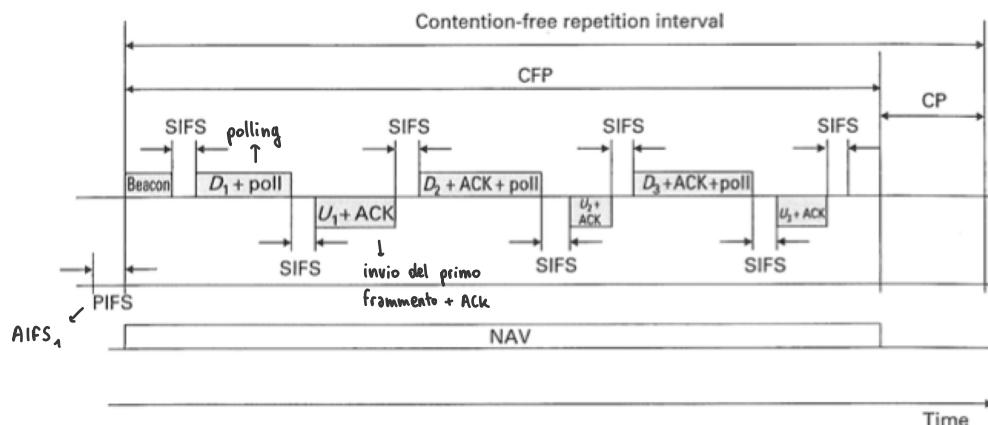
Un meccanismo opzionale di RTS/CTS utilizza il NAV per impedire che i terminali spediscano frame contemporaneamente a terminali nascosti. Un terminale A invia un frame RTS per richiedere ad una stazione B il permesso di spedirgli un frame: se B riceve la richiesta, risponde con un frame CTS per indicare che il canale è libero, A a sua volta spedisce il frame e fa partire un timer per l'acknowledgment, che gli verrà poi inviato da B. Se A non riceve l'acknowledgment, viene ipotizzata una collisione e ricomincerà tutto di nuovo dopo un backoff. Se B è occupato invece, aggiornerà il proprio NAV così che tutte le stazioni possano rendersene conto. RTS/CTS sembra efficace, ma non lo è con i frame corti e con gli access point, oltre che rallentare molto le operazioni.



• Differenza tra switch e router

La differenza fondamentale tra un router e uno switch è che un router collega fra loro reti diverse, mentre uno switch connette più dispositivi per creare una rete. Il router è un dispositivo utilizzato per l'internetworking: viene utilizzato per connettere LAN o WAN indipendenti insieme; ha un indirizzo fisico e logico per ciascuna delle sue interfacce. Il router funziona sul livello fisico, collegamento dati e livello di rete, mentre lo switch funziona solo a livello dati e rete. Lo scopo principale del router è determinare il percorso migliore per un pacchetto per raggiungere la destinazione, d'altra parte uno switch riceve un pacchetto e lo elabora per determinare l'indirizzo di destinazione e inoltra il pacchetto per indirizzare l'indirizzo di destinazione rivelato.

- Se una stazione deve trasmettere 10 pacchetti e ne ha trasmesso 1, ha la priorità sulla trasmissione degli altri 9? NO, se sono frammenti è un discorso, pacchetti è un altro - spiegare figura sotto



La figura sopra descrive l'invio di diversi frammenti in 802.11: i frammenti sono numerati individualmente e soggetti ad acknowledgement tramite un protocollo stop and wait; una volta che il controllo del canale è stato acquisito, molteplici frammenti vengono spediti in una rapida sequenza. Partono uno dopo l'altro con in mezzo un ack (e forse ritrasmissioni) finché l'intero frame non sia stato spedito con successo o sia stato raggiunto il tempo massimo di trasmissione. Il NAV rende le altre stazioni inattive solo fino al successivo ack, per questo un altro meccanismo tutela i frammenti estendo il CSMA/CA con intervalli scelti con cura tra i frame. L'intervallo tra frame di dati è chiamato DIFS (*CF interframe spacing*), l'intervallo più breve è SIFS (*short interframe spacing*) e viene usato per permettere a coloro che partecipano a uno scambio di messaggi di poter partire per primi. Ci sono quindi intervalli che definiscono diversi livelli di priorità, come gli AIFS con quattro diversi livelli di priorità, e poi intervalli EIFS utilizzati solo dalle stazioni che hanno ricevuto un frame difettoso o sconosciuto per segnalare il problema. Nella figura è mostrato un normale scambio di frammenti, senza problemi o collisioni (no EIFS), in cui ogni frammento è intervallato da SIFS e si ripete l'operazione di polling dopo ogni invio di frammento + ACK; il tutto descritto nel tempo dal NAV.

• Livello 3: chi instrada i pacchetti?

Il router. Ogni router ha una tabella interna che indica dove devono essere inviati i pacchetti diretti a ogni possibile destinazione. Ogni voce della tabella è costituita da una coppia di valori, che rappresentano una destinazione e la linea di trasmissione da utilizzare per raggiungerla. Possono essere utilizzate solo linee collegate direttamente. L'algoritmo che gestisce la tabella e prende le decisioni di instradamento è chiamato algoritmo di routing; non è garantita la consegna ordinata: se un router rileva un ingorgo nel traffico, può cambiare la traiettoria di un pacchetto, e rimandare il suo invio. L'idea alla base dei circuiti virtuali è quella di evitare di dover scegliere una nuova strada per ogni pacchetto inviato. Invece, quando è stabilita una connessione, il percorso dal computer sorgente a quello di destinazione viene scelto durante l'impostazione della connessione e archiviato nelle tabelle dei router. Quel percorso è utilizzato per tutto il traffico che scorre attraverso la connessione, con funzionamento identico a quello del sistema telefonico. Quando la connessione viene rilasciata, anche il circuito virtuale viene terminato. Con un servizio orientato alla connessione, ogni pacchetto contiene un identificatore che indica il circuito virtuale di appartenenza.

Tabelle di routing: se voglio mandare un pacchetto ad un dispositivo non collegato alla rete locale, devo mandare l'indirizzo IP ad un router locale che lo manderà ad un altro router ecc..

Ogni router "chiede" quali sono i router vicini, riempie una tabella e la manda in broadcast. Ciò si ripete per ogni router, così che il router stesso sia in grado di ricreare la topologia della rete. Fa ciò con un algoritmo chiamato DISTRA, con cui trova il percorso a lunghezza minima.

• **Cos'è la tabella ARP, protocollo ARP, anche da wireshark**

La tabella ARP presente su ogni computer deve mantenere la corrispondenza tra indirizzo IP e indirizzo MAC di ogni scheda di rete (può essere anche impostata in maniera statica, anche se non è molto comune). Viene popolata tramite un protocollo ARP (*Address Resolution Protocol*) che si colloca sopra il livello 2 ma non usa ancora il protocollo IP, un protocollo ausiliario che diciamo "sta a metà", si appoggia direttamente ai frame. Il protocollo ARP appartiene alla suite del protocollo IPv4 e il suo compito è fornire la "mappatura" tra l'indirizzo IP (4 byte) e l'indirizzo MAC (6 byte) corrispondente di un terminale. Il suo analogo in IPv6 è *Neighbor Discovery Protocol* o NDP.

Per inviare un pacchetto IP ad un calcolatore della stessa sottorete, è necessario incapsularlo in un pacchetto di livello datalink, che dovrà avere come indirizzo destinazione il MAC Address del calcolatore a cui lo si vuole inviare. ARP viene utilizzato per ottenere questo indirizzo. Se il pacchetto deve essere inviato ad un calcolatore di un'altra sottorete, ARP viene utilizzato per scoprire il MAC Address del gateway o del router. In ogni calcolatore il protocollo ARP tiene traccia delle risposte precedentemente ottenute in una apposita cache (ARP cache) per evitare di dover utilizzare continuamente ARP prima di inviare ciascun pacchetto. Le informazioni contenute nella cache vengono cancellate tipicamente dopo 5 minuti. Funzionamento: l'host che vuole conoscere il MAC address di un altro host, di cui conosce l'indirizzo IP, invia in broadcast una richiesta ARP contenente il proprio indirizzo MAC e l'indirizzo IP del destinatario di cui si vuole conoscere il MAC address. Tutti i calcolatori della sottorete ricevono la richiesta: in ciascuno di essi il protocollo ARP verifica, confrontando l'IP proprio con quello inviato, se viene richiesto il proprio MAC address. L'host di destinazione che riconoscerà il proprio indirizzo IP nel pacchetto di ARP-request, provvederà ad inviare una risposta (ARP reply) contenente il proprio MAC direttamente all'host mittente (quindi in unicast). In questo modo, ogni host può scoprire l'indirizzo fisico degli altri host sulla stessa sottorete. Questo è particolarmente importante nel caso si vogliano conoscere i dispositivi di rete quali gateway, router per verificare eventuali malfunzionamenti di un nodo della rete stessa.

L'arrivo dell'ARP-request ad un nodo aggiorna completamente la tabella ARP presente nella cache a lei dedicata dal protocollo, senza rispetto per le voci preesistenti nella tabella di Routing.

Gli ARP request/reply sono 28 byte incapsulati all'interno di un frame Ethernet. L'ethernet destination address viene impostato a ff:ff:ff:ff:ff:ff (broadcast per indirizzi ARP) se viene effettuata un'ARP request. In questi 28 byte è contenuto il campo ARP operation, impostato a 1 se si tratta di una richiesta ARP, a 2 se si tratta di una risposta.

Protocollo ARP → se non trova gli indirizzi → richiesta: lo manda in broadcast sulla rete locale → manda ARP reply con tabella
↓
di 4 campi (IP/MAC
sorgente e destinatario)
riempita.
risolvere IP: trovare indirizzo MAC corrispondente

Protocollo inverso: RARP

arp&PING(ICMP).dump

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

Aplica un filtro di visualizzazione ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:aa:bc:59:c6:2b	Broadcast	ARP	42	Who has 10.0.1.2? Tell 10.0.1.1 → lui manda in broadcast la richiesta
2	0.021324	00:aa:f6:fd:9c:a3	00:aa:bc:59:c6:2b	ARP	60	10.0.1.2 is at 00:aa:f6:fd:9c:a3 → risposta
3	0.021355	10.0.1.1	10.0.1.2	ICMP	98	Echo (ping) request id=0x01f2, seq=1/256, ttl=64 (reply in 4)
4	0.043686	10.0.1.2	10.0.1.1	ICMP	98	Echo (ping) reply id=0x01f2, seq=1/256, ttl=64 (request in 3)
5	1.0001554	10.0.1.1	10.0.1.2	ICMP	98	Echo (ping) request id=0x01f2, seq=2/512, ttl=64 (reply in 6)
6	1.022876	10.0.1.2	10.0.1.1	ICMP	98	Echo (ping) reply id=0x01f2, seq=2/512, ttl=64 (request in 5)
7	2.002841	10.0.1.1	10.0.1.2	ICMP	98	Echo (ping) request id=0x01f2, seq=3/768, ttl=64 (reply in 8)
8	2.024102	10.0.1.2	10.0.1.1	ICMP	98	Echo (ping) reply id=0x01f2, seq=3/768, ttl=64 (request in 7)
9	5.241304	00:aa:f6:fd:9c:a3	00:aa:bc:59:c6:2b	ARP	60	Who has 10.0.1.1? Tell 10.0.1.2 → chiede l'indirizzo Ethernet del 10.0.1.1
10	5.241336	00:aa:bc:59:c6:2b	00:aa:f6:fd:9c:a3	ARP	42	10.0.1.1 is at 00:aa:bc:59:c6:2b

↳ il protocollo ARP qui probabilmente deve completare la tabella

```
> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
> Ethernet II, Src: 00:aa:bc:59:c6:2b (00:aa:bc:59:c6:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)
```

0000 ff ff ff ff ff ff 00 aa bc 59 c6 2b 08 06 00 01 ·Y·+···
0010 08 00 06 04 00 01 00 aa bc 59 c6 2b 0a 00 01 01 ·Y·+···
0020 00 00 00 00 00 00 0a 00 01 02

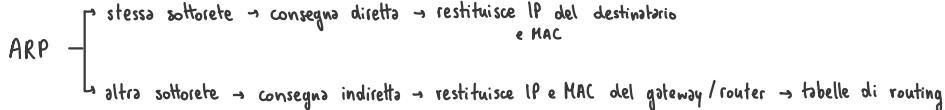
arp&PING(ICMP).dump | Pacchetti: 10 - visualizzati: 10 (100.0%)

ICMP: Internet Control Message Protocol è un protocollo di servizio per reti a pacchetto che si occupa di trasmettere informazioni riguardanti malfunzionamenti, informazioni di controllo o messaggi tra i vari componenti di una rete di calcolatori. È encapsulato direttamente in IP e viene utilizzato da molti applicativi di rete, tra cui ping.

Ping: Packet Internet Groper è un utility di amministrazione per reti di computer usata per misurare il tempo, espresso in millisecondi, impiegato da uno o più pacchetti ICMP a raggiungere un dispositivo di rete e a ritornare indietro all'origine. È prettamente utilizzato per verificare la presenza e la raggiungibilità di un altro computer connesso in rete e per misurare le latenze di trasmissione di rete.

• Consegnare indirettamente in Internet, come funziona la consegna di un pacchetto IP nella stessa sottorete

Estratto l'indirizzo IP dal pacchetto (datagramma IP), mettendo in AND l'indirizzo IP e la subnet-mask si ottiene l'indirizzo di Net_Id che individua la sottorete di destinazione: se tale indirizzo coincide con quello dell'interfaccia del router si ha intradamento diretto all'interno della stessa sottorete dopo risoluzione IP/MAC tramite ARP, se invece la Net_Id calcolata non corrisponde a nessuna interfaccia del router verso le rispettive sottoreti interconnesse, dopo l'indirizzamento ottenuto tramite consultazione della *tavola di routing*, si ha intradamento indiretto ovvero il router utilizza la consegna diretta (locale) di livello 2 tramite protocollo ARP per inviare il pacchetto verso il gateway o il successivo router (next hop) il quale procederà per successivi salti di intradamento fino alla sottorete di destinazione (Host_Id) sempre seguendo le informazioni contenute nelle tabelle di routing dei rispettivi router, a loro volta create e aggiornate dinamicamente dai protocolli di routing.



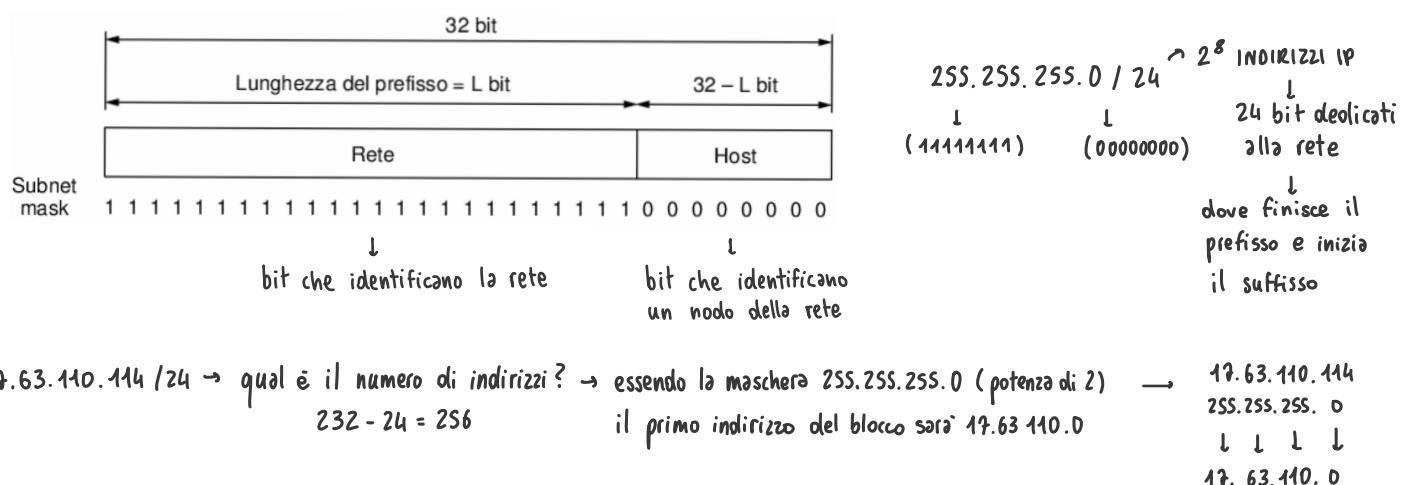
• Protocollo IP: quante versioni e le loro differenze

Internet Protocol (IP) è un protocollo di rete che si occupa di indirizzamento/instradamento, appartenente alla suite di protocolli Internet TCP/IP. Classificato al livello 3 del modello ISO/OSI, è un protocollo a pacchetti senza connessione e di tipo best effort, che non garantisce cioè alcuna forma di affidabilità della comunicazione in termini di controllo di errore, controllo di flusso e controllo di congestione, di cui si occupano i protocolli di trasporto di livello superiore come TCP. È importante ricordare che l'indirizzo IP non è parte del computer, ma si riferisce alla connessione che si instaura, infatti un dispositivo può avere tanti indirizzi IP quante connessioni (solitamente un host ne ha uno, ma ad esempio un router ne ha tanti). Gli indirizzi IP hanno validità topologica, dicono dove si trova il dispositivo o router connesso.

Il principale compito di IP è l'indirizzamento tra sottoreti eterogenee, che a livello locale utilizzano invece un indirizzamento proprio tipicamente basato sull'indirizzo fisico o MAC e protocolli di livello datalink (2). La "colla" tra queste sottoreti è il protocollo IP: ciò comporta l'assegnazione a ciascun terminale che ne fa richiesta di un nuovo diverso indirizzo, univocamente associato all'indirizzo MAC locale, detto indirizzo IP (tramite protocolli RARP, BOOTP o DHCP); e la definizione delle modalità tese a individuare il percorso di rete per interconnettere due qualunque sottoreti durante una comunicazione tra host sorgente di una certa sottorete e host destinatario di un'altra sottorete, cui l'indirizzo IP appartiene.

Correntemente sono usate due versioni del protocollo IP, l'originaria versione 4 e la più recente versione 6, nata dall'esigenza di gestire meglio il crescente numero di dispositivi connessi ad internet.

L'IPv4 (*Internet protocol version 4*) è un protocollo di rete che permette l'instradamento delle connessioni, ed è tutt'oggi il protocollo più diffuso per il collegamento a Internet. Ogni dispositivo connesso a una rete ha bisogno di un indirizzo IP per essere individuato e per comunicare con gli altri dispositivi: IPv4 permette di assegnare a ogni dispositivo un indirizzo univoco a 32 bit (4 byte), scritti in notazione decimale puntata, in cui ognuno dei 4 byte è rappresentato in forma decimale con un numero che varia tra 0 e 255. Ogni indirizzo a 32 bit è composto da una parte di rete di lunghezza variabile nei primi bit è da una parte per l'host negli ultimi. La parte di rete ha lo stesso valore per tutti gli host di una singola rete, ed è chiamata prefisso. La dimensione è determinata dal numero di bit nella porzione di rete; i bit rimanenti nella parte dedicata all'host possono variare. Questo significa che la dimensione deve essere una potenza di due e per convenzione viene scritta dopo il prefisso dell'indirizzo IP come / (slash) seguito dalla lunghezza in bit della parte dedicata alla rete. A volte i prefissi sono descritti semplicemente dalla loro lunghezza, come in "/16". La lunghezza del prefisso corrisponde a una maschera binaria di 1 nella parte di rete. Quando viene scritta in questo modo è chiamata subnet mask (*maschera di sottorete*).



Indirizzi speciali IPv4: 0.0.0.0 → utilizzato dagli host al momento del boot e significa questa rete o questo host;

tutti 1 o 255.255.255.255 → tutti gli host di una rete, trasmissione broadcast locale; forma 127 → controlli all'interno dell'host stesso (loopback);

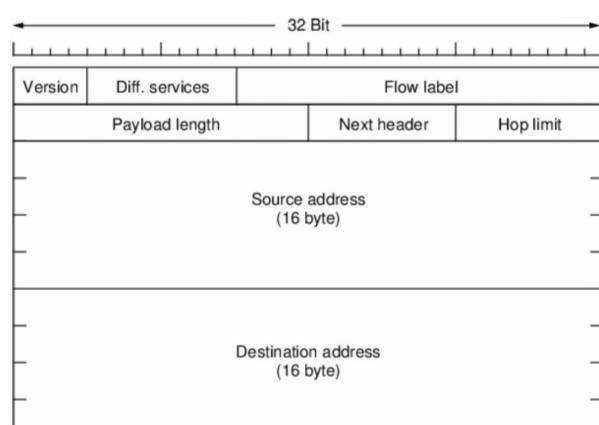
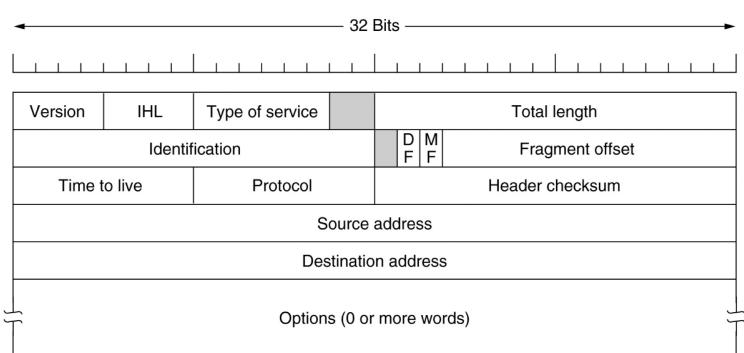
Molto presto gli indirizzi a disposizione di IPv4 iniziarono a scarseggiare, fino a terminare: per risolvere il problema, l'unica soluzione a lungo termine fu quella di adottare indirizzi più grandi; IPv6 è un progetto che usa indirizzi a 128 bit. È un protocollo a livello di rete differente, che non interagisce con IPv4 nonostante le similarità. L'intestazione di IPv6 differisce innanzitutto da IPv4 per il campo *Version*: è sempre di 6 bit per IPv6 (e 4 per IPv4). I router sono in grado di esaminare questo campo per determinare il tipo di pacchetto. Per scrivere gli indirizzi a 16 byte è stata scelta una nuova notazione. Gli indirizzi sono scritti come otto gruppi di quattro cifre esadecimale separate da due punti:

8000:0000:0000:0000:0123:4567:89AB:CDEF

importante da ricordare: servivano indirizzi più grandi (e più veloci) quindi niente Checksum, frammentazione diversa + Flow label

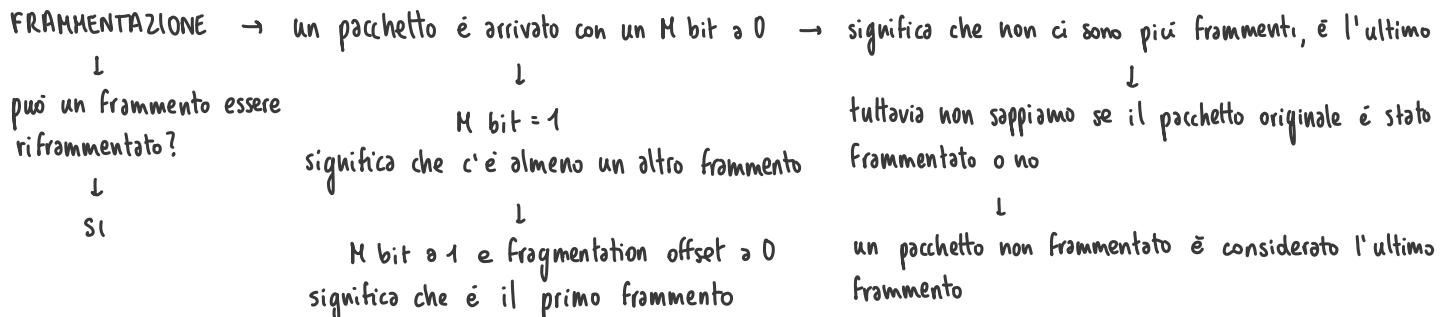
La differenza tra questi due formati di indirizzi sta nel fatto che gli indirizzi IPv6 sono più lunghi e formattati in modo diverso, quindi esistono più configurazioni di indirizzi IPv6 univoci possibili.

IPv4	IPv6
Gli indirizzi IPv4 sono lunghi 32 bit	Gli indirizzi IPv6 sono lunghi 128 bit
Indirizzi composti da numeri binari rappresentati in decimali (192.168.5.18)	Indirizzi composti da numeri binari rappresentati in esadecimale (50b2:6400::6c3a:b17d:0:10a9)
La frammentazione è fatta dal mittente e inoltrata router	La frammentazione è fatta solo dal mittente
Nessuna identificazione del flusso di pacchetti	Identificazione del flusso di pacchetti disponibile all'interno dell'intestazione utilizzando il campo Flow Label
È presente un campo Checksum	Nessun campo Checksum presente
Campi Opzioni sono disponibili in IPv4	Non ci sono campi di opzione, ma sono disponibili IPv6 Extension (estensioni dell'intestazione)
Si usa ARP per mappare indirizzi IPv4 a indirizzi MAC	ARP viene sostituito con Neighbor Discovery Protocol (NDP)
I messaggi broadcast sono disponibili	I messaggi broadcast non sono disponibili. Viene invece usato un Link-local scope per il multicast (FF02 :: 1)
Configurazione manuale (statico) di indirizzi IPv4 o DHCP (dinamica) necessaria per configurare gli indirizzi	Configurazione automatica degli indirizzi
4.3 bilioni di indirizzi: necessità di riusare e mascherare gli indirizzi (NAT)	7.9 x 10^28 indirizzi: ogni dispositivo può averne uno unico



• Frammentazione in IPv4

Nell'intestazione IPv4 sono presenti 3 campi dedicati alla frammentazione: il campo di 1 bit *DF “don't fragment”* rappresenta un ordine che impone ai router di non dividere in frammenti il datagramma; un altro campo di 1 bit, *MF “more fragments”* è impostato a 1 per tutti i frammenti tranne l'ultimo, per comunicare quando sono arrivati tutti; infine il campo *Fragment offset*, di 13 bit, indica la posizione del frammento nel datagramma corrente. Tutti i frammenti di un datagramma tranne l'ultimo devono essere multipli di 8 byte.



• Differenza indirizzi pubblici/privati

Indirizzi pubblici → visibili su Internet, univoci in tutto il mondo, assegnati da qualcuno
Indirizzi privati → non visibili su Internet, visibilità limitata alla rete locale: nessun pacchetto contenente questi indirizzi può apparire su internet, tre intervalli riservati:

10.0.0.0 – 10.255.255.255/8 (16.777.216 host)

172.16.0.0 – 172.31.255.255/12 (1.048.576 host)

192.168.0.0 – 192.168.255.255/16 (65.536 host)

• Come funziona il NAT (forma scritta)

Nonostante la soluzione a lungo termine per risolvere la questione dell'esaurimento degli indirizzi IP sia che tutta Internet passi a IPv6, e ci vorranno anni prima che questa transizione si completi, serviva una soluzione sul breve termine. La soluzione adottata si chiama NAT (*Network Address Translation*): l'idea di base è assegnare a ogni azienda o casa un singolo indirizzo IP (o, al massimo, un piccolo numero di indirizzi) per il traffico di Internet. Dentro la rete del cliente, ogni computer riceve un indirizzo IP unico, utilizzato per instradare il traffico interno alla rete locale. Tuttavia quando un pacchetto sta per lasciare la rete locale per dirigersi verso l'ISP (internet service provider) viene eseguita una traduzione di indirizzo dall'unico indirizzo IP interno a quello pubblico condiviso. Utilizza i 3 intervalli di indirizzi privati.

Ad esempio, dentro i locali dell'azienda, ogni macchina ha un unico indirizzo espresso nella forma 10.x.y.z, ma quando un pacchetto lascia la rete locale, passa attraverso un apparato NAT che converte l'indirizzo IP interno (10.0.0.1) nel vero indirizzo IP assegnato all'azienda (198.60.42.12 ad esempio). L'apparato NAT è spesso abbinato a un firewall, inserito all'interno di un singolo apparecchio che protegge la rete locale, controllando attentamente tutti i dati che entrano e che escono dalla LAN.

In che modo l'apparato NAT sceglie l'indirizzo di destinazione interno? La maggior parte dei pacchetti IP trasporta al suo interno TCP o UDP, entrambi protocolli che hanno intestazioni che contengono una porta sorgente e una porta di destinazione. Le porte sono numeri interi lunghi 16 bit che indicano dove inizia e finisce la connessione e offrono il campo che permette di far funzionare NAT.

Quando un processo desidera stabilire una connessione TCP con un processo remoto, si lega a una porta TCP inutilizzata sulla sua macchina; questa porta è chiamata porta sorgente e indica al codice di TCP dove devono essere inviati i pacchetti in arrivo appartenenti alla connessione. Il processo fornisce anche una porta di destinazione che indica chi deve ricevere i pacchetti sulla parte remota. Le porte da 0 a 1023 sono riservate a servizi noti; per esempio, la porta 80 è utilizzata dai server Web, e ciò consente a client remoti di individuare il servizio. Ogni volta che un pacchetto diretto verso l'esterno raggiunge l'apparato NAT, l'indirizzo sorgente 10.x.y.z è sostituito dall'indirizzo IP dell'azienda. Inoltre, il campo *Source port* è sostituito da un indice che punta alla tabella di traduzione da 65.536 voci dell'apparato NAT. Questa voce della tabella contiene l'indirizzo IP originale e la porta sorgente originale. Infine i codici di controllo delle intestazioni IP e TCP sono ricalcolati e inseriti nel pacchetto. È necessario sostituire il campo *Source port* perché, per esempio, due connessioni distinte dalle macchine 10.0.0.1 e 10.0.0.2 possono entrambe utilizzare la porta 5000; perciò la porta sorgente non basta, da sola, a identificare il processo che sta trasmettendo.

Quando un pacchetto trasmesso dall'ISP raggiunge l'apparato NAT, la *Source port* nell'intestazione TCP viene estratta e utilizzata come indice nella tabella di associazione dell'apparato. Dalla voce individuata, l'apparato estrapola l'indirizzo IP interno e la *Source port* TCP originale, quindi inserisce entrambe le informazioni nel pacchetto; poi ricalcola e inserisce nel pacchetto sia il checksum IP sia quello TCP. Alla fine il pacchetto viene passato al router interno per il normale inoltro basato sull'indirizzo 10.x.y.z.

• Come viene deciso l'indirizzo IP del pc all'accensione: DHCP

DHCP sta per Dynamic Host Configuration Protocol (protocollo di configurazione IP dinamica) ed indica un protocollo ausiliario che permette ai dispositivi o terminali di una certa rete locale di ricevere automaticamente ad ogni richiesta di accesso, da una rete IP, la configurazione IP necessaria per stabilire una connessione e operare su una rete più ampia. Esso consente l'interoperabilità con tutte le altre sotto-reti scambiandosi dati, purché anch'esse integrate allo stesso modo con il protocollo IP. Infatti in una rete basata sul protocollo IP, ogni calcolatore ha bisogno di un indirizzo IP, scelto in modo tale che appartenga all'insieme di indirizzi possibili assegnati all'intera sottorete (cioè al Net_ID) a cui è collegato e che sia univoco, cioè non ci siano altri calcolatori che stiano già utilizzando quell'indirizzo. Il compito di assegnare manualmente gli indirizzi IP ai calcolatori comporta infatti un rilevante onere per gli amministratori di rete, soprattutto in reti di grandi dimensioni o in caso di numerosi computer che si connettono a rotazione solo a ore o giorni determinati. Inoltre gli indirizzi IPv4 disponibili (attualmente usati nella quasi totalità delle reti al mondo) con l'aumentare dei computer connessi a Internet hanno cominciato a scarseggiare, diminuendo la disponibilità di IP fissi per eventuali configurazioni statiche. DHCP supporta questo compito automaticamente e in maniera dinamica, cioè solo quando richiesto dall'host.

client DHCP → calcolatore che ha bisogno di ottenere un indirizzo IP valido per la sottorete a cui è collegato

server DHCP → calcolatore che assegna gli indirizzi IP

DHCP relay → calcolatore che si occupa di inoltrare le richieste DHCP ad un server, qualora questo non sia sulla stessa sottorete

Allocazione dinamica: allocazione automatica di indirizzi temporanei "affitto";

Allocazione automatica: il server assegna automaticamente un indirizzo IP a un client richiedente nell'intervallo definito dall'amministratore;

Allocazione statica: il server emette un indirizzo IP privato dipendente dal MAC address di ciascun client, basato su una mappatura predefinita da parte dell'amministratore.

• TCP: caratteristiche

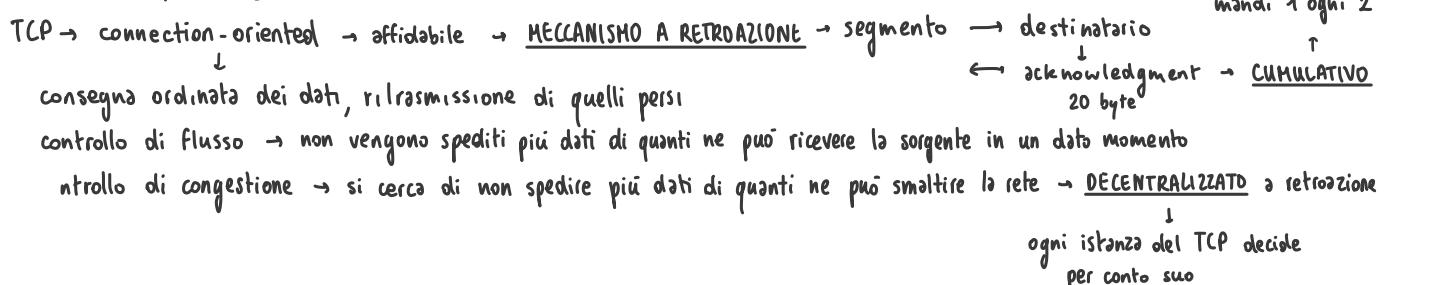
TCP è stato progettato appositamente per fornire un flusso di byte affidabile end-to-end su una internetwork affidabile. È molto più complesso dell'UDP in quanto deve offrire un servizio orientato alla connessione affidabile, a partire dal servizio datagram offerto da IP. I dati sono visti come un flusso di byte, anche se sono organizzati in pacchetti chiamati segmenti.

Il servizio TCP è ottenuto con la creazione di punti terminali di un sistema di comunicazione da parte di mittente e ricevente, chiamati socket. Ogni socket possiede un numero (indirizzo) composto dall'indirizzo IP dell'host e da un numero di 16 bit locale all'host, chiamato porta. Per ottenere il servizio TCP si deve stabilire esplicitamente una connessione tra una socket su una macchina e una socket su un'altra macchina. Una socket può essere usata per più connessioni contemporaneamente. I numeri di porta minori di 1024 sono riservati per servizi standard, e prendono il nome di *well-known port*.

Tutte le connessioni TCP sono di tipo full-duplex punto a punto. Full-duplex indica che il traffico può procedere in entrambe le direzioni contemporaneamente. Punto a punto significa che ogni connessione ha esattamente due punti terminali. TCP non supporta il multicast o il broadcast. Una connessione TCP è un flusso di byte, non un flusso di messaggi. I confini dei messaggi non vengono conservati da un'estremità all'altra della connessione. Per esempio, se il processo mittente segue quattro invii da 512 byte su un flusso TCP, questi dati possono essere consegnati al processo ricevente come quattro blocchi di 512 byte, due blocchi di 1024 byte, un solo blocco di 2048 byte o in altri modi ancora.

Nel TCP i dati inviati (in byte), organizzati in segmenti, devono essere confermati da un ACK (acknowledgment): il TCP è bidirezionale, e gli ACK possono sfruttare i segmenti dati della connessione di ritorno. Funzionamento: ogni segmento inviato è contraddistinto da un numero di sequenza che indica il primo byte del payload; ogni ACK indica quale è il primo byte che il ricevitore si aspetta di ricevere e implicitamente conferma tutti i byte precedenti (ACK cumulativo); se un segmento è perso il ricevitore all'arrivo dei segmenti successivi invia degli ACK duplicati (DupACK). Ogni segmento ricevuto genera l'invio di un ACK (esclusa delayed ACK per i quali si ha un ACK ogni due); il tempo che intercorre fra l'invio di un segmento e l'arrivo dell'ACK da lui causato si chiama *RTT (Round Trip Time)*. È il tempo dell'anello di retroazione del TCP. Più è grande, peggio funziona il TCP.

IP → connection-less → unreliable

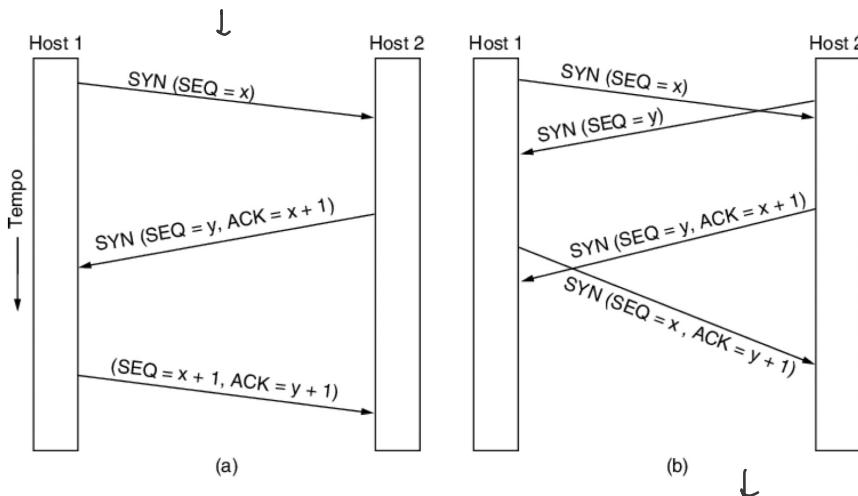


Una funzionalità vitale di TCP, che guida la struttura del protocollo, consiste nel fatto che ogni byte in una connessione TCP ha un proprio numero di sequenza a 32 bit. Le entità TCP di invio e ricezione scambiano i dati sotto forma di segmenti. Un segmento TCP consiste di un'intestazione fissa di 20 byte (più una parte facoltativa) seguita da zero o più byte di dati. Il software TCP decide la dimensione dei segmenti e può accumulare in un segmento i dati provenienti da più invii oppure divider i dati di un invio in più segmenti.

Il protocollo di base utilizzato dalle entità TCP è il protocollo a finestra scorrevole con una dimensione dinamica della finestra. Quando un mittente trasmette un segmento, avvia anche un timer. Quando il segmento arriva a destinazione, l'entità TCP ricevente invia un segmento (con i dati, se esistono, oppure senza) contrassegnato da un numero di acknowledgement uguale al numero di sequenza successivo che prevede di ricevere la dimensione della finestra disponibile. Se il timer del mittente scade prima della ricezione dell'acknowledgement, questi ritrasmette il segmento.

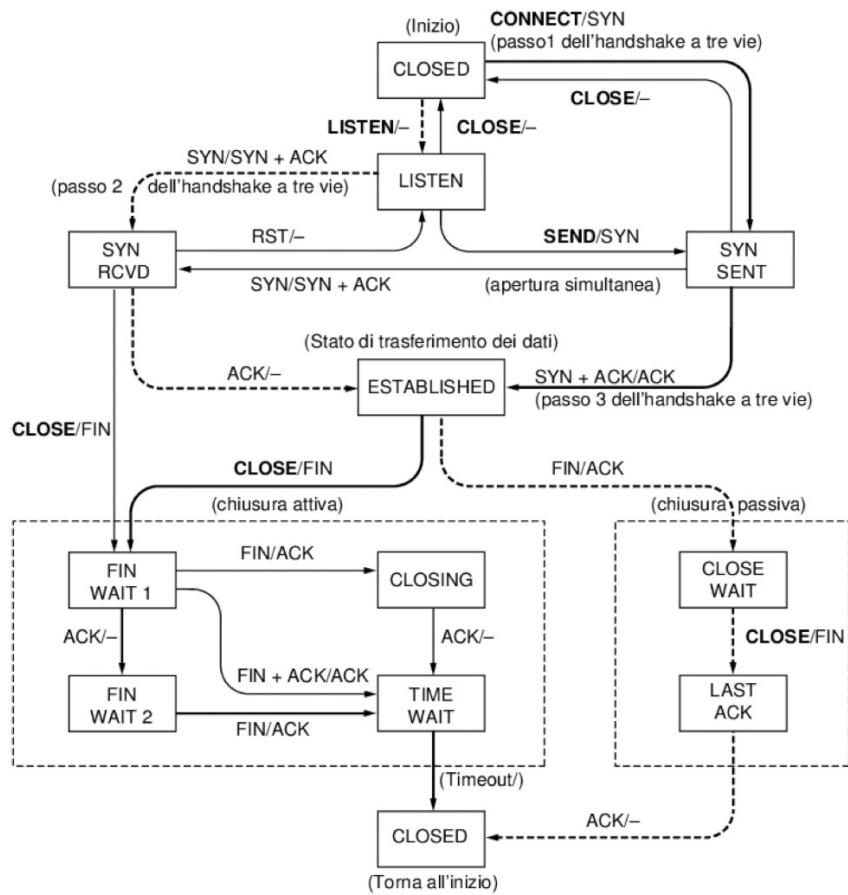
• TCP: come si instaura una connessione?

Per stabilire una connessione, un lato (per esempio il server) attende in modo passivo una connessione in ingresso eseguendo nell'ordine le primitive LISTEN e ACCEPT che possono indicare una sorgente specifica oppure nessuna in particolare. L'altro lato (diciamo il client) segue una primitiva CONNECT, specificando l'indirizzo IP e la porta a cui vuole connettersi, la dimensione massima del segmento TCP che è intenzionato ad accettare e, facoltativamente, alcuni dati utente. La primitiva CONNECT invia un segmento TCP con il bit SYN a 1 e il bit ACK a 0, poi attende una risposta. Quando questo segmento arriva a destinazione, l'entità TCP controlla se esiste un processo che ha eseguito una LISTEN sulla porta indicata nel campo Destination port e in caso negativo invia una risposta con il bit RST a 1 per rifiutare la connessione. Se un processo è in ascolto sulla porta, gli viene dato il segmento TCP in ingresso; quindi può accettare o rifiutare la connessione. Se accetta, viene restituito al mittente un segmento di acknowledgement. Si noti che un segmento SYN consuma un byte nello spazio delle sequenze, così da poter ricevere un acknowledgement non ambiguo.



Nel caso che due host tentino contemporaneamente di stabilire una connessione tra le stesse due socket, il risultato di questi eventi è la costituzione di una sola connessione, non due, perché le connessioni sono identificate dai loro punti terminali. Se sia la prima attivazione che la seconda generano una connessione identificata da (x, y) , viene comunque creata una sola voce nella tabella per (x, y) . Si ricordi che il numero di sequenza iniziale, scelto da ciascun host, dovrebbe variare lentamente invece di essere una costante, come 0. Questa regola serve a proteggere dai pacchetti in ritardo e duplicati.

Ogni connessione inizia nello stato CLOSED. Lo stato viene lasciato quando la connessione esegue un'apertura passiva (LISTEN) o un'apertura attiva (CONNECT). Se l'altro lato esegue l'operazione opposta, viene stabilita una connessione e lo stato diventa ESTABLISHED. Il rilascio della connessione può essere iniziato da una qualunque delle parti. Una volta completato, lo stato torna a essere CLOSED.



Client = linea continua spessa; Server = linea tratteggiata spessa.

Quando un programma applicativo sulla macchina client genera una richiesta CONNECT, l'entità TCP locale crea un record per la connessione, la contrassegna nello stato SYN SENT e invia un segmento SYN. Si noti che molte connessioni potrebbero essere aperte (o venire aperte) contemporaneamente da più applicazioni, pertanto lo stato è relativo alla singola connessione ed è registrato nel relativo record. Quando arriva il SYN + ACK, TCP invia l'ACK finale dell'handshake a tre vie e passa nello stato ESTABLISHED. Ora è possibile inviare e ricevere i dati. Dal punto di vista del server. Il server segue una LISTEN e attende per vedere che cosa succede. Quando al server arriva un SYN, questo manda un acknowledgement e passa allo stato SYN RCVD; nel momento in cui anche il SYN del server riceve un acknowledgement, l'handshake a tre vie è completo e il server passa allo stato ESTABLISHED.

• TCP: fase di chiusura

Quando un'applicazione ha finito, segue una primitiva CLOSE che provoca l'invio da parte dell'entità TCP locale di un segmento FIN e l'attesa dell'ACK corrispondente (il riquadro tratteggiato indicato come "chiusura attiva"). Quando arriva l'ACK, viene eseguita una transizione allo stato FIN WAIT 2 e una direzione della connessione si chiude. Quando anche l'altro lato esegue la chiusura, viene ricevuto un FIN che richiede l'invio di un acknowledgment. Ora entrambi i lati sono chiusi, ma TCP attende un tempo uguale al doppio del tempo di vita massimo del pacchetto per garantire che tutti i pacchetti della connessione siano scaduti (nel caso in cui l'acknowledgement fosse andato perso). Alla cadenza del timer, TCP elimina il record della connessione. Quando il client ha terminato di trasmettere i propri dati effettua una CLOSE, che provoca l'invio da parte del client di un FIN al server (il riquadro tratteggiato indicato come "chiusura passiva"). Il server, ricevuto il segnale FIN, segue a sua volta una CLOSE; operazione che prevede l'invio di un FIN al client. All'arrivo dell'acknowledgement relativo a quest'ultimo FIN da parte del client, il server rilascia la connessione e ne elimina il relativo record.

• TCP: controllo di flusso / receiver window o advertised window

Per il controllo di flusso il protocollo TCP utilizza un protocollo a finestre scorrevoli. Ciò che il mittente può spedire dipende non solo dagli ack ricevuti ma anche dallo spazio disponibile nel buffer del ricevente. Lo spazio disponibile nel buffer è indicato dal campo window size che la stazione di destinazione manda al mittente in ogni segmento in modo che il mittente possa regolare la trasmissione dei segmenti; la dimensione della finestra indica il numero di byte che il destinatario è in grado di ricevere.

Quando la dimensione della finestra è 0 il mittente non può spedire dati; può comunque sempre inviare un segmento con dati urgenti, per esempio per far terminare il processo sull'altra stazione; inoltre può spedire un segmento di un byte per fare in modo che l'altra stazione ripeta qual è il prossimo byte atteso e segnali la dimensione della finestra; questo segmento sonda (che prende il nome di window probe) viene mandato quando scade un apposito timer per evitare problemi nel caso fosse stata segnalata una finestra di dimensione 0 e poi fosse andato perduto il segmento con l'aggiornamento della dimensione della finestra.

Quando trasmette un segmento il mittente fa partire un timer; se entro lo scadere del timer non riceve la conferma del destinatario ritrasmette il segmento.

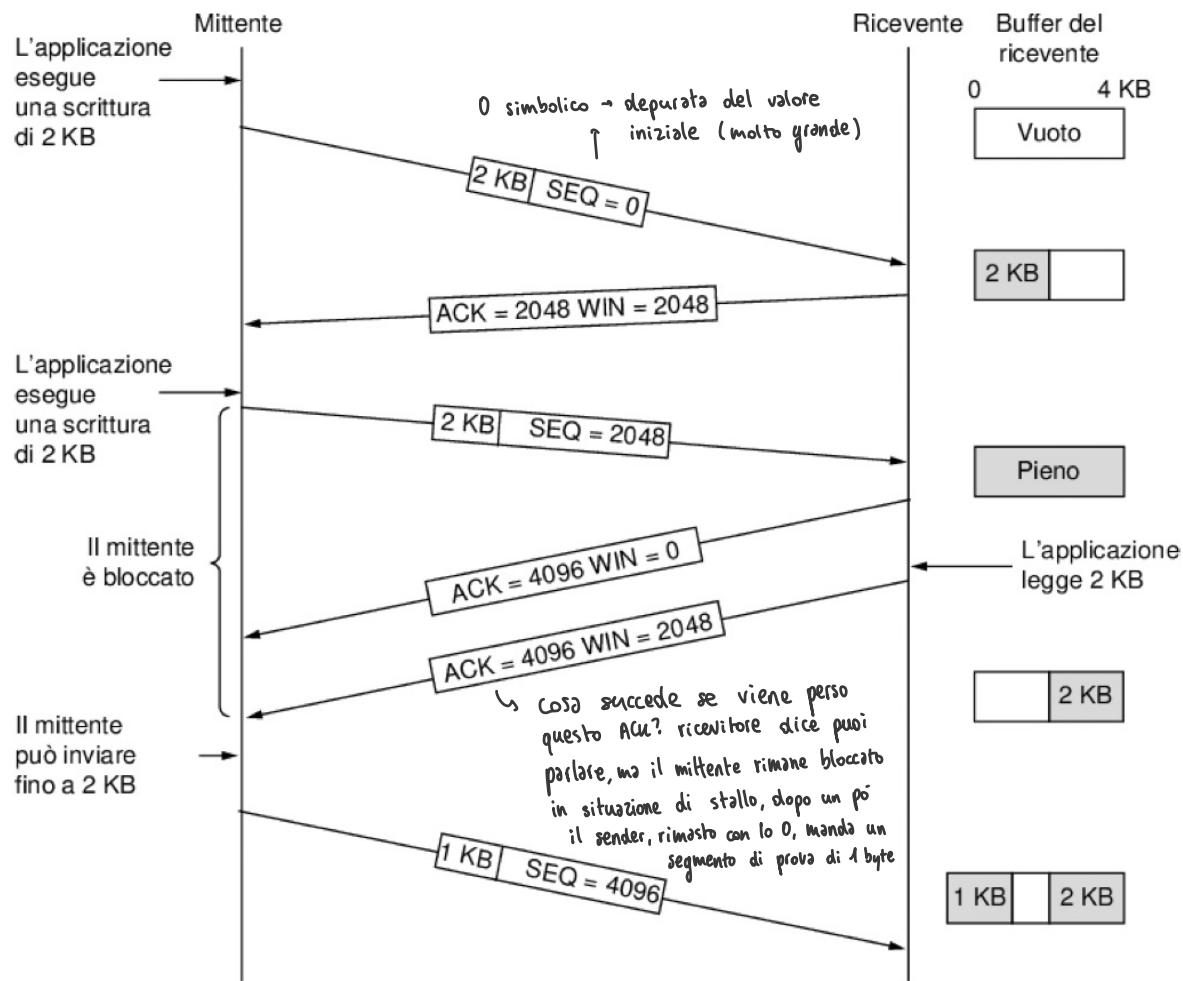
La durata del timer è importante perché se è troppo grande ci saranno ritardi ogni volta che viene perso un pacchetto e se è troppo piccola ci saranno molte ritrasmissioni non necessarie. Il timer dovrebbe essere uguale al tempo di andata e ritorno dalla destinazione, ma questo tempo è molto variabile e dipende anche dalle congestioni della rete.

Il timer viene impostato dinamicamente in base a un algoritmo (Jacobson 1988) che misura continuamente le prestazioni della rete; ma se un segmento viene ritrasmesso perché il timer è scaduto la durata del timer viene raddoppiata.

Non è necessario che il mittente trasmetta i dati non appena arrivano dall'applicazione e non è nemmeno richiesto che il ricevente spedisca immediatamente la conferma dei dati ricevuti. Vengono utilizzati dei metodi per rendere più efficiente la trasmissione.

Il destinatario, invece di inviare la conferma per ogni segmento ricevuto, può aspettare di ricevere alcuni segmenti e confermare solo l'ultimo byte ricevuto; per evitare che ciò faccia scadere il timer del mittente anche il destinatario usa un timer, allo scadere del quale deve inviare una conferma al mittente

Si supponga per esempio che il ricevente abbia un buffer di 4096 byte. Se il mittente trasmette un segmento di 2048 byte che viene correttamente ricevuto, il ricevente darà un acknowledgment al segmento. Tuttavia, dal momento che ora possiede solo 2048 byte di spazio nel buffer (fino a quando l'applicazione non rimuove dei dati), "pubblicizza" una finestra di 2048 byte partendo dal successivo byte atteso. Ora il mittente trasmette altri 2048 byte che ricevono acknowledgment; la finestra pubblicizzata ha però dimensione 0. Il mittente deve fermarsi finché il processo applicativo sull'host ricevente non abbia rimosso dati dal buffer; in quel momento TCP potrà pubblicizzare una finestra più grande e potranno essere spediti altri dati.



• TCP: controllo di congestione / congestion window

Quando il carico applicato a qualsiasi rete è più di quello che questa riesce a gestire, si crea una congestione, e Internet non fa eccezione. Il livello di rete rileva la congestione quando le code sui router aumentano e cerca di gestirla scartando dei pacchetti. È compito del livello di trasporto ricevere i segnali di congestione dal livello di rete e rallentare il tasso del traffico inviato sulla rete.

Il controllo della congestione di TCP utilizza una finestra e interpreta la perdita di pacchetti come segnali binari di congestione. Per farlo, TCP mantiene una finestra di congestione la cui dimensione è pari al numero di byte che il mittente può avere sulla rete in qualsiasi momento. Il tasso corrispondente è la dimensione della finestra diviso per il round-trip time della connessione. TCP aggiusta la dimensione della finestra. Si ricordi che la finestra di congestione viene mantenuta in aggiunta alla finestra di controllo del flusso che specifica il numero di byte che il destinatario può inserire nel suo buffer. Entrambe le finestre vengono seguite in parallelo e il numero di byte che possono essere spediti è il più piccolo delle due finestre. TCP smetterà di spedire dati se la finestra di controllo di congestione o quella di controllo di flusso sono temporaneamente piene.

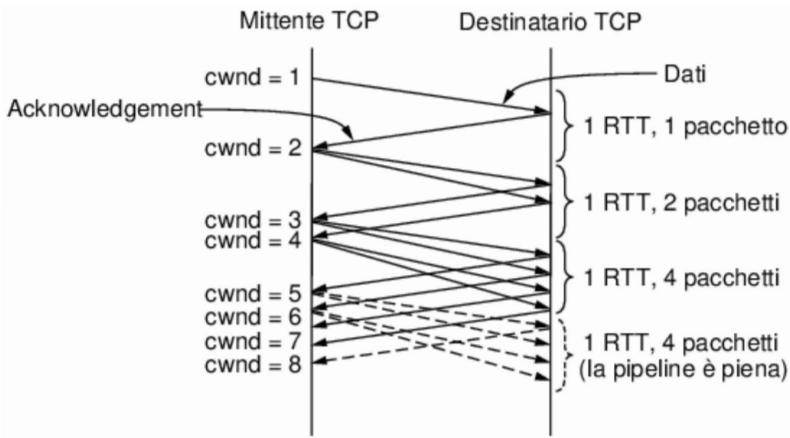
Tutti gli algoritmi di TCP usati su Internet assumono che i pacchetti persi siano causati da una congestione; si affidano ai timeout analogamente ai minatori con i canarini. È necessario un buon timer di ritrasmissione a rilevare i segnali di perdita di pacchetti in modo preciso e tempestivo. Dato un buon timeout di ritrasmissione, il mittente TCP può tracciare il numero di byte in sospeso che occupa la rete, cercando semplicemente la differenza tra le sequenze numeriche trasmesse e quelle che hanno ricevuto acknowledgement.

Analizziamo per esempio cosa accade quando un mittente su una rete veloce (la linea a 1 Gbps) spedisce un gruppo di quattro pacchetti molto ravvicinati tra loro a un destinatario su una rete lenta (la linea a 1 Mbps) che è il collo di bottiglia, o la parte più lenta, sul percorso. Inizialmente i quattro pacchetti viaggiano sul collegamento alla velocità a cui possono essere spediti dal mittente. Nel router vengono accodati durante l'inoltro perché ci vuole più tempo per spedire un pacchetto su un collegamento lento rispetto che per ricevere il pacchetto successivo su un collegamento veloce. Ma la coda non è lunga, perché solo un piccolo numero di pacchetti è stato spedito in una volta. Da notare la lunghezza maggiore dei pacchetti sul collegamento lento: lo stesso pacchetto di 1 KB ora è più lungo perché si impiega più tempo per spedirlo.

Alla fine i pacchetti arrivano al destinatario, dove verrà generato un acknowledgement. Le tempistiche degli acknowledgement riflettono i tempi impiegati da pacchetti per arrivare al destinatario dopo aver attraversato il collegamento più lento e risultano sparpagliati se confrontati ai pacchetti originali sul collegamento veloce. Nell'attraversare la rete per tornare al mittente gli acknowledgement mantengono inalterate queste tempistiche. Gli acknowledgement tornano al mittente a circa lo stesso tasso a cui i pacchetti possono essere spediti sul collegamento più lento lungo il percorso; questo è esattamente il tasso che il mittente desidera usare. Se inserisce nuovi pacchetti nella rete con questo tasso, essi saranno spediti alla velocità consentita dal collegamento più lento, ma non saranno messi in coda e non congestioneranno alcun router lungo il percorso. Questa temporizzazione è conosciuta come ack clock ed è una parte essenziale di TCP. Usando un ack clock, TCP rende più regolare il traffico ed evita code inutili a router.

Quando viene stabilita una connessione, il mittente inizializza la finestra di congestione a un valore iniziale basso, di al massimo quattro segmenti; i dettagli sono descritti nell'RFC 3390, l'uso di quattro segmenti è un incremento, sulla base dell'esperienza, da un primo valore iniziale di un solo segmento. Il mittente, quindi, spedisce la finestra iniziale. I pacchetti impiegheranno un round-trip time per ottenere i relativi acknowledgement. Per ogni segmento che ha ricevuto 'acknowledgement prima dello scadere del timer di ritrasmissione il mittente aggiunge alla finestra di congestione il valore in byte di un segmento; in più, quando quel segmento ha ricevuto un acknowledgement ci sarà un segmento in meno in rete. Il risultato è che ogni acknowledgement, permette a due segmenti di essere inviati. La finestra di congestione raddoppia a ogni round-trip time. Questo algoritmo è chiamato **slow start** (partenza lenta): nel primo round-trip il mittente immette un pacchetto nella rete (e il destinatario riceve un pacchetto), due pacchetti sono spediti nel successivo round-trip time e quindi quattro nel terzo round-trip time.

Lo slow start funziona bene su una gamma di velocità e round-trip time e usa un ack clock per allineare il tasso delle trasmissioni del mittente al percorso di rete. Si guardi il modo in cui gli acknowledgement ritornano dal mittente al destinatario nella figura sotto. Quando un mittente riceve un acknowledgement aumenta la finestra di congestione di un'unità e immediatamente spedisce due pacchetti sulla rete: un pacchetto è per l'incremento, l'altro è un rimpiazzo per il pacchetto il cui acknowledgement è stato ricevuto; in ogni momento il numero di pacchetti senza acknowledgement è dato dalla finestra di congestione. Tuttavia questi due pacchetti non arriveranno necessariamente a destinazione così ravvicinati come quando sono stati spediti; Se il percorso nella rete è lento, gli acknowledgement arrivano lentamente (dopo un ritardo di un RTT). Se il percorso di rete è veloce, gli acknowledgement arrivano velocemente (ancora, dopo un RTT). Tutto ciò che il mittente deve fare è seguire la tempistica dell'ack clock per inviare nuovi pacchetti, che è ciò che fa lo slow start.



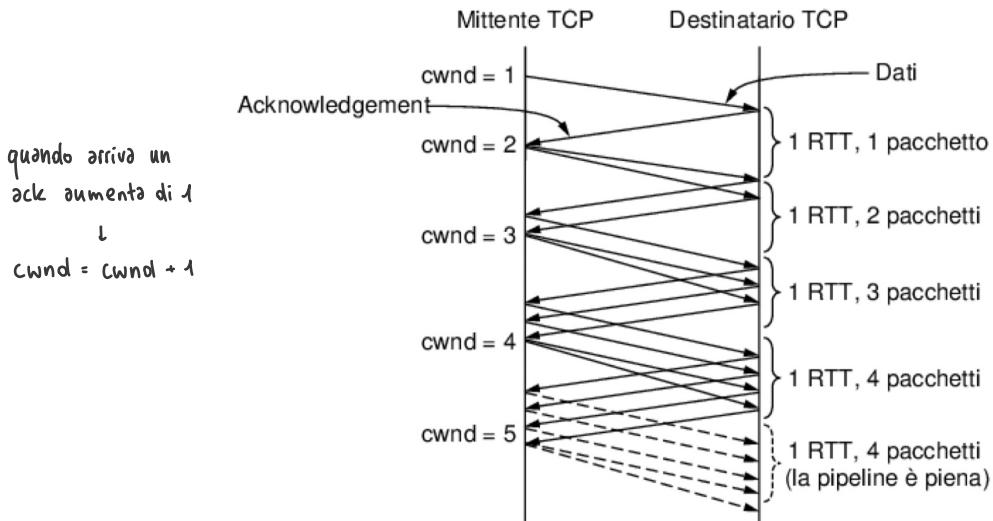
Poiché lo slow start causa una crescita esponenziale, prima o poi (più prima che poi) spedirà in rete troppi pacchetti e troppo velocemente. Quando ciò avviene, si creano code nella rete e quando le code saranno piene uno o più pacchetti saranno persi. Dopo che questo accade, il mittente TCP andrà in timeout quando un acknowledgement non riuscirà ad arrivare in tempo. Nella figura viene data evidenza del fatto che lo slow start cresce troppo velocemente; dopo due RTT, ci sono quattro pacchetti in rete. Questi quattro pacchetti impiegano un intero RTT per arrivare al destinatario, quindi una finestra di quattro pacchetti ha la giusta dimensione per questa connessione. Tuttavia, non appena vengono ricevuti gli acknowledgement di questi pacchetti, lo slow start continua a far crescere la finestra di congestione, raggiungendo gli otto pacchetti in un altro RTT. Solo quattro di questi pacchetti possono raggiungere il destinatario in un RTT, indipendentemente da quanti ne vengono spediti. In altre parole, la pipeline di rete è piena.

Ulteriori pacchetti spediti in rete dal mittente si accumuleranno nelle code sui router, in quanto non potranno essere consegnati al destinatario abbastanza velocemente. Per tenere sotto controllo lo slow start, il mittente mantiene una soglia per il collegamento chiamata slow start threshold (soglia di slow start). Inizialmente questo valore è fissato, in modo arbitrariamente elevato, alla dimensione della finestra di controllo di flusso, in maniera tale da non limitare la connessione. TCP continua a far crescere la finestra di congestione nello slow start finché non si verifica un timeout o la finestra di congestione supera la soglia (o si è riempita la finestra del destinatario).

Ogni volta che viene rilevata la perdita di un pacchetto, per esempio attraverso un timeout, la soglia di slow start è fissata per essere la metà della finestra di congestione e l'intero processo ricomincia. L'idea è che la finestra corrente sia troppo grande perché ha già causato una congestione che solo ora viene rilevata da un timeout. Metà della finestra, usata con successo in un momento precedente, è probabilmente una stima migliore per una finestra di congestione vicina alla capacità del percorso, ma che non causerà perdite. Nel nostro esempio la crescita della finestra di congestione a otto pacchetti può causare perdite, mentre la finestra di congestione di quattro pacchetti nel RTT precedente era il valore corretto.

Ogni volta che viene oltrepassata la soglia dello slow start, TCP passa dallo slow start all'**incremento additivo**; in questa modalità, la finestra di congestione viene aumentata di un segmento per ogni round-trip time. Questo, però, viene solitamente implementato con un aumento (più piccolo) per ogni segmento, per cui si riceve un acknowledgement invece di crescere a ogni RTT, come nel caso dello slow start. Indichiamo la finestra di congestione con $cwnd$ e la grandezza massima di segmento con MSS . Un'approssimazione di uso comune consiste nell'aumentare $cwnd$ di $(MSS \times MSS)/cwnd$ per ognuno dei $cwnd/MSS$ pacchetti che potrebbero ricevere un acknowledgement.

Alla fine di ogni RTT la finestra di congestione del mittente è cresciuta abbastanza per poter introdurre un pacchetto aggiuntivo in rete. Rispetto allo slow start, il tasso lineare di crescita è molto più lento. Non fa molta differenza per piccole finestre di congestione, come in questo caso, ma ne fa molta sul tempo impiegato per far crescere la finestra di congestione a 100 segmenti, ad esempio.



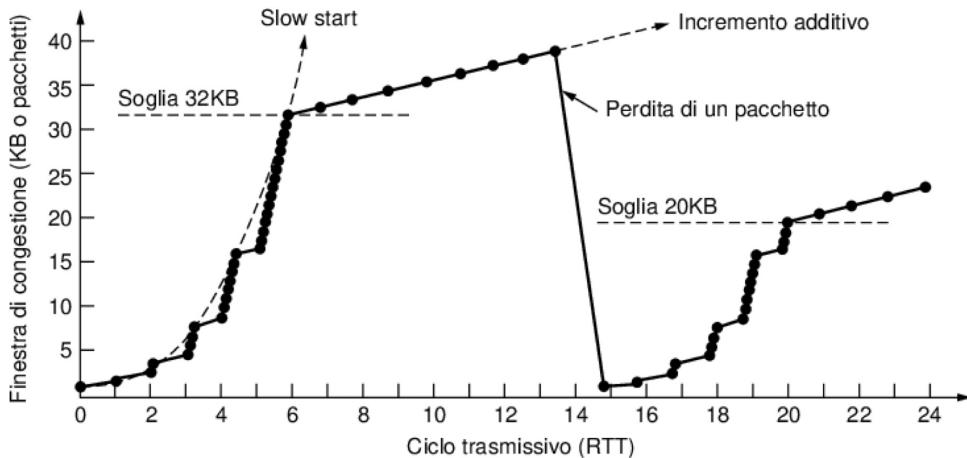
Dopo che viene perso un pacchetto il destinatario non può mandare acknowledgement per quelli successivi, quindi il numero di acknowledgement rimarrà fisso e il mittente non sarà in grado di spedire alcun nuovo pacchetto sulla rete, perché la sua finestra di congestione rimarrà piena. Questa condizione può continuare per un periodo relativamente lungo finché scatta il timer e il pacchetto perso viene ritrasmesso. A questo punto TCP parte nuovamente con lo slow start.

Il mittente ha un modo veloce per riconoscere che uno dei suoi pacchetti è stato perso. Non appena i pacchetti successivi al pacchetto perduto arrivano a destinazione, innescano degli acknowledgement che ritornano al mittente. Questi acknowledgement portano lo stesso numero e sono chiamati duplicate acknowledgement (dupACK). Ogni volta che il mittente riceve un acknowledgement duplicato è probabile che un altro pacchetto sia arrivato al destinatario e il pacchetto perso non si sia ancora fatto vivo.

Poiché i pacchetti possono prendere diversi percorsi attraverso rete, possono arrivare fuori ordine generando acknowledgement duplicati anche se nessun pacchetto è stato perso. Tuttavia, nella maggior parte dei casi, questo non è comune su Internet; anche quando c'è un riordino dei pacchetti a causa di percorsi diversi, i pacchetti ricevuti non sono troppo fuori ordine. Quindi TCP assume in modo abbastanza arbitrario che tre acknowledgement duplicati implicino che un pacchetto sia stato perso e l'identità del pacchetto perso si possa facilmente dedurre dal numero negli acknowledgement: è il prossimo pacchetto nella sequenza. Questo pacchetto può essere ritrasmesso immediatamente, prima che scatti il timeout di ritrasmissione.

Tale meccanismo prende il nome di **fast retransmission**. Quando entra in funzione, la soglia di slow start viene di nuovo portata a metà della corrente finestra di congestione, proprio come con un timeout. Lo slow start può essere fatto ripartire portando la finestra di congestione a un pacchetto. Con questa dimensione della finestra, dopo il round-trip time necessario per mandare un acknowledgement relativo al pacchetto ritrasmesso, verrà spedito un nuovo pacchetto insieme a tutti i dati spediti prima del rilevamento della perdita.

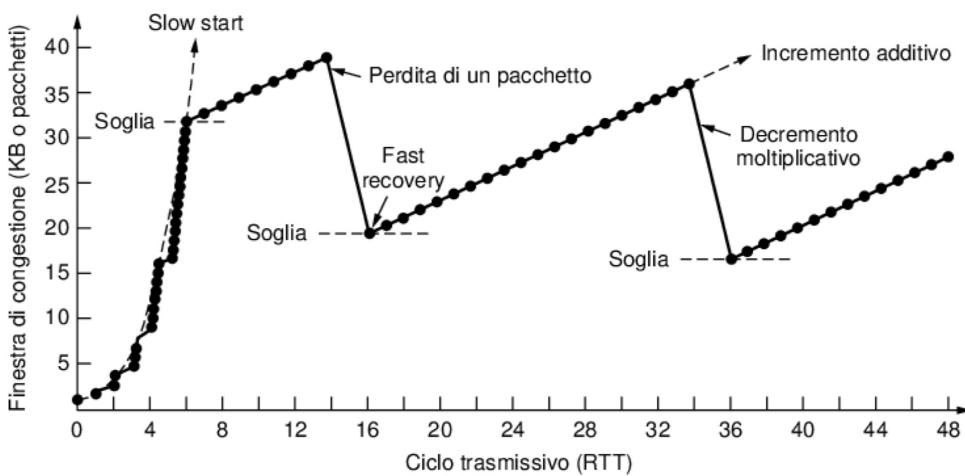
La dimensione massima di un segmento nell'esempio è di 1 KB. All'inizio la finestra di congestione era 64 KB, ma si era verificato un timeout, quindi la soglia fu impostata a 32 KB e la finestra di congestione a 1 KB per la trasmissione O. La finestra di congestione cresce esponenzialmente finché raggiunge la soglia (32 KB). Invece che in maniera continua, la finestra è incrementata ogni volta che arriva un acknowledgement, il che porta a un profilo con una serie di gradini. Superata la soglia la finestra aumenta linearmente e viene incrementata di un segmento ogni RTT.



Al tredicesimo ciclo una trasmissione si perde nella rete. Questo fatto viene rilevato quando arrivano tre acknowledgement duplicati. In quel momento il pacchetto perso viene ritrasmesso, la soglia è portata a metà della finestra corrente (finora 40 KB, quindi la metà è 20 KB) e lo slow start comincia di nuovo, da capo.

TCP Tahoe (che include buoni timer di ritrasmissione) fornisce un algoritmo di controllo della congestione funzionante che risolve il problema del collasso per congestione.

Il **fast recovery** è una modalità temporanea che cerca di mantenere l'ack clock in funzione con una finestra di congestione grande quanto la nuova soglia o nel momento della ritrasmissione veloce quanto metà del valore della finestra di congestione. Si rimane in modalità temporanea finché non ci si accorge che il numero di pacchetti in rete è sceso al di sotto della nuova soglia contando gli acknowledgement duplicati (inclusi i tre che hanno fatto scattare la ritrasmissione veloce); questa procedura impiega circa metà del round-trip time. Da quel momento in poi, un nuovo pacchetto può essere spedito per ogni acknowledgement duplicato ricevuto. Trascorso un round-trip time dopo la ritrasmissione veloce il pacchetto perduto avrà ricevuto un acknowledgement. A quel punto il flusso di acknowledgement duplicati cesserà e la modalità di fast recovery sarà abbandonata. La finestra di congestione sarà impostata alla nuova soglia di slow start e aumenterà con crescita lineare.



Il risultato di questo comportamento è che TCP evita lo slow start, tranne all'inizio della connessione e quando si verifica un timeout. Quest'ultimo può ancora verificarsi quando va perso più di un pacchetto e la ritrasmissione veloce non recupera adeguatamente. Invece di slow start multipli, la finestra di congestione della connessione in atto segue un profilo a dente di sega per gli incrementi additivi (di un segmento ogni RTT) e i decrementi moltiplicativi (di metà in un RTT).

TCP Reno è essenzialmente TCP Tahoe con fast recovery. Dopo uno slow start iniziale, la finestra di congestione aumenta linearmente finché viene rilevata la perdita di un pacchetto per via di acknowledgement duplicati. Il pacchetto perso è ritrasmesso e il fast recovery è usato per mantenere attivo ack clock finché la ritrasmissione riceve un acknowledgement. A quel punto la finestra di congestione è riavviata dalla nuova soglia di slow start, invece che da 1. Questo comportamento continua all'infinito e la connessione passa la maggior parte del tempo con la sua finestra di congestione vicina al valore ottimale del prodotto banda-ritardo.

La versione TCP NewReno (1996) opera un avanzamento parziale del numero di acknowledgement dopo una ritrasmissione per trovare e rimediare a un'altra perdita.

FINESTRE E VELOCITÀ DI TRASMISSIONE

↓
W: numero massimo di segmenti che possono essere spediti dopo l'ultimo confermato

↓
Se voglio $Tx \cdot \text{banda}$ disponibile, la finestra dovrà essere uguale al prodotto banda ritardo (BDP) $\rightarrow W = B \cdot RTT$

$$W = \min(cwnd, rwnd) \rightarrow Tx = \min\left(\frac{cwnd}{RTT}, \frac{rwnd}{RTT}\right)$$

↓
massima Tx ammessa dal controllo di congestione

↓
massima Tx ammessa dal controllo di flusso

]} La velocità di trasmissione può quindi essere limitata dal controllo di flusso o da quello di congestione

↓
Sia cwnd che rwnd cambiano nel tempo

PERDITE → si ritiene sia andato perso un segmento se

scade l'RTO (time-out) → ovvero se non sono arrivati ACK non duplicati per un periodo più lungo di RTO (RTO è calcolato dinamicamente sulla base di RTT)

arrivano 3 dupACK

! sono generati alla ricezione dei segmenti successivi a quello mancante

↓ aggiunta da TCP Tahoe

↓ RTT: round-trip time → tempi di arrivo degli ACK

era l'unica condizione nelle prime versioni di TCP

• Finestra di congestione a R4, cosa accade dopo 3 dup-ack?

TCP Tahoe -> finestra di congestione dimezzata a 1 MSS (massima dimensione segmento TCP) -> slow start

TCP Reno -> fast retransmit

• Timer nel TCP

1) Timer di ritrasmissione (RTO): serve a verificare che ciascun segmento trasmesso venga riscontrato. La corretta impostazione di questo timer è difficile, in quanto un timer troppo breve comporta ritrasmissioni inutili, mentre uno troppo lungo comporta attese in caso di effettiva perdita di pacchetti. Tale intervallo ovviamente almeno pari all'RTT (tempo di percorrenza a due vie di un pacchetto per tornare al mittente sotto forma di ACK), tipicamente variabile in maniera aleatoria. TCP allora aggiusta continuamente il timer basandosi su una stima a media mobile del Round Trip Time.

2) Timer di persistenza: il TCP utilizza il metodo della finestra scorrevole per gestire il controllo di flusso. Tra i valori validi di questo campo vi è anche lo zero, a significare che il ricevente richiede l'interruzione momentanea dell'invio di dati. Nel caso in cui il pacchetto che riapre la finestra venga perso, il mittente del canale TCP rimarrà però in attesa indefinita. Per evitare questo, il TCP avvia il timer di persistenza ogni volta il ricevente chiude la finestra. Quando questo timer scade, il mittente invia un pacchetto sonda al ricevente, provocandone una risposta: in questo modo il mittente potrà essere sicuro che la finestra sia chiusa (riceve window = 0) o sbloccarsi dallo stallo (window != 0).

3) Timer di keepalive: quando non ci sono dati da trasmettere sulla connessione, alcune implementazioni di TCP consentono di trasmettere periodicamente segmenti vuoti, detti keepalive, per evitare di mantenere indefiniteamente in memoria connessioni con sistemi che potrebbero non essere più attivi. Quando si usano i keepalive, è dunque presente questo timer che li fa trasmettere, con un valore tipico di due ore.

4) Timed wait: prima di disconnettere effettivamente una connessione, i due estremi del canale attendono un tempo pari al doppio del tempo di vita di un comune pacchetto; questo evita che dei pacchetti possano rimanere circolanti per la rete dopo la chiusura.

• TCP/UDP

Le principali differenze tra TCP e UDP (User Datagram Protocol), sono:

- TCP è un protocollo orientato alla connessione. Pertanto per stabilire, mantenere e chiudere una connessione è necessario inviare segmenti di servizio i quali aumentano l'overhead di comunicazione. Al contrario, UDP è senza connessione ed invia solo i datagrammi richiesti dal livello applicativo; (segmenti = TCP, datagrammi = UDP);
- UDP non offre nessuna garanzia sull'affidabilità della comunicazione, ovvero sull'effettivo arrivo dei segmenti, né sul loro ordine in sequenza di arrivo. Al contrario il TCP, tramite i meccanismi di acknowledgement e ritrasmissione su timeout, riesce a garantire la consegna dei dati, anche se al costo di un maggiore overhead. TCP riesce altresì a riordinare i segmenti in arrivo presso il destinatario attraverso il campo sequence number del suo header;
- L'oggetto della comunicazione di TCP è un flusso di byte, mentre quello di UDP sono singoli datagrammi.

L'utilizzo del protocollo TCP rispetto a UDP è, in generale, preferito quando è necessario avere garanzie sulla consegna dei dati o sull'ordine di arrivo dei vari segmenti (come per esempio nel caso di trasferimenti di file). Al contrario UDP viene principalmente usato quando l'interazione tra i due host è idempotente o nel caso si abbiano forti vincoli sulla velocità e l'economia di risorse della rete (es. streaming in tempo reale, videogiochi multiplayer).

• DTN, e differenze con TCP/IP

TCP e molti altri protocolli di trasporto sono basati sull'ipotesi che mittente e destinatario siano continuamente connessi da qualche percorso funzionante, altrimenti il protocollo fallisce e i dati non possono essere recapitati. In alcune reti spesso un percorso end-to-end non esiste. Un esempio è una rete spaziale come i satelliti LEO (low-earth orbit) che entrano ed escono dal raggio di azione delle stazioni a terra.

In queste reti occasionalmente connesse i dati possono ancora essere comunicati immagazzinandoli nei nodi e inoltrandoli più tardi quando c'è un collegamento funzionante: alla fine i dati saranno inviati alla destinazione. Una rete la cui architettura è basata su questo approccio è chiamata DTN (delay-tolerant network o disruption-tolerant network, rete con tolleranza di ritardo o rete con tolleranza alle rotture).

Il lavoro sulle DTN iniziò nel 2002 quando IETF organizzò un gruppo di ricerca sull'argomento. L'ispirazione per le DTN proveniva dagli sforzi per inviare pacchetti nello spazio: le reti spaziali devono avere a che fare con comunicazioni intermittenti e ritardi molto lunghi. Kevin Fall osservò che le idee per queste Internet Interplanetaria potevano essere applicate alle reti sulla Terra in cui la connettività intermittente era la norma.

Alcuni esempi di situazioni in cui conveniva introdurre la DTN sono: navicelle spaziali, reti militari, una specie di "risposta ai disastri" (terremoti ecc), reti subacquee, sensori, reti nei paesi in via di sviluppo

Ovviamente non si colloca all'interno del modello ISO/OSI, è una nuova architettura, un livello "aggiunto".

L'architettura DTN si basa sull'introduzione del livello Bundle tra il livello trasporto e applicazione. I bundle sono grandi pacchetti di dati in questo livello, che adopera un'architettura store-and-forward (un bundle viene ricevuto, conservato e poi trasmesso quando è possibile).

- I protocolli di trasporto in questi segmenti sono specializzati nell'occuparsi dei ritardi per cui sono progettati queste reti;
- I dati devono essere immagazzinati nei nodi intermedi (applicazioni "data mule");

Immagazzinare i dati può essere utile per gestire le mancanze di un path ent-to-end: ciò può avvenire per connessioni con interruzioni casuali (terra) o per connessioni con interruzioni conosciute (spazio); si riesce ad avere una ripresa anche con lunghi RTT.

L'immagazzinamento e l'inoltro dei bundle da parte dei nodi DTN suona simile all'accodamento (nei buffer) e l'inoltro dei pacchetti nei router, ma ci sono differenze qualitative. Nei router in Internet gli accodamenti avvengono per millisecondi o al massimo secondi, mentre nei nodi DTN i bundle possono essere memorizzati per ore, finché un autobus non arriva in città, mentre un aeroplano completa il volo, finché un sensore non raccoglie abbastanza energia solare per funzionare, finché un computer non si sveglia e così via. Questi esempi indicano una seconda differenza, cioè che i nodi potrebbero muoversi (con l'autobus o l'aereo) mentre tengono memorizzati dei dati, e questo movimento può anche essere una parte importante nella consegna dei dati. I router in Internet non possono muoversi.

Note utili per la risoluzione degli esercizi

Formule per i numeri complessi	
Modulo	$r = z := \sqrt{a^2 + b^2} \quad r \geq 0$
Argomento	$\theta := \operatorname{Arg}(z) \in [0, 2\pi)$

$$\rightarrow \theta := \operatorname{Arg}(z) = \begin{cases} \frac{\pi}{2} & \text{se } a = 0, b > 0 \\ \frac{3\pi}{2} & \text{se } a = 0, b < 0 \\ \text{non definito} & \text{se } a = 0, b = 0 \\ \arctan\left(\frac{b}{a}\right) & \text{se } a > 0, b \geq 0 \\ \arctan\left(\frac{b}{a}\right) + 2\pi & \text{se } a > 0, b < 0 \\ \arctan\left(\frac{b}{a}\right) + \pi & \text{se } a < 0, b \text{ qualsiasi} \end{cases}$$

Componenti circuituali		
Condensatore	C	$\frac{1}{j\omega C}$
Resistenza	R	R
Induttore	L	$j\omega L$

IN SERIE \rightarrow SOMMA DEUE IMPEDENZE

IN PARALLELO \rightarrow SOMMA DEGLI INVERSI DEUE IMPEDENZE

$$H(\omega) = \frac{\dot{V}_y}{\dot{V}_x} = \frac{Z_2}{Z_1 + Z_2} \quad \dot{V}_x = I(Z_1 + Z_2) \quad \dot{V}_y = IZ_2$$

PER PASSARE DA PULSAZIONI A FREQUENZE \rightarrow 1) SOSTITUIRE GLI ω CON $2\pi f$
2) MOLTIPLICARE PER 2π

$$\text{PULSAZIONE DI RISONANZA: } \omega_0 = \frac{1}{\sqrt{LC}}$$

$$\text{PULSAZIONE DI TAGLIO: } \omega_t = \frac{1}{RC}$$

RISPOSTA A SEGNAI SVILUPPABILI IN SERIE DI FOURIER:

$$X(\omega)$$

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

$$Y(\omega) = X(\omega)H(\omega)$$

$$y(t) = \sum_{n=-\infty}^{\infty} c_n H(n\omega_0) e^{jn\omega_0 t}$$

$$x(t) = A_o + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t - \varphi_n) \quad y(t) = A_o H(0) + \sum_{n=1}^{\infty} A_n T(n\omega_0) \cos[n\omega_0 t - \varphi_n - \beta(n\omega_0)]$$

$$x(t) = \int_0^{+\infty} V(\omega) \cos[\omega t - \varphi(\omega)] d\omega \quad y(t) = \int_0^{+\infty} V(\omega) T(\omega) \cos[\omega t - \varphi(\omega) - \beta(\omega)] d\omega$$

$$\text{TRASFORMATA DI FOURIER: } X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$$

$$\text{SPECTRO DI AMPIEZZA MONOLATERO: } V(\omega) = \frac{|X(\omega)|}{\pi}$$

$$\text{ANTITRASFORMATA DI FOURIER: } x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} dt$$

$$\text{SPECTRO DI FASE MONOLATERO: } \Phi(\omega) = -\arg\{X(\omega)\}$$

$$\text{INTEGRALE DI FOURIER: } x(t) = \int_0^{+\infty} V(\omega) \cos[\omega t - \Phi(\omega)] d\omega$$

$$\text{FUNZIONE DI TRASFERIMENTO RETE LINEARE: } H(\omega) = F[h(t)] \rightarrow Y(\omega) = X(\omega)H(\omega)$$

$$\text{CARATTERISTICHE DI AMPIEZZA E FASE: } \begin{cases} T(\omega) = |H(\omega)| \\ \beta(\omega) = -\arg\{H(\omega)\} \end{cases}$$

$$\text{RISPOSTA A UNA SINUSOIDA: } x(t) = A_x \cos(\omega_x t - \varphi_x)$$

IMPULSO RETTANGOLARE \leftrightarrow sinc

$$y(t) = A_y \cos(\omega_x t - \varphi_x)$$

$$\text{POTENZA ISTANTANEA: } p(t) = x^*(t)x(t) = |\dot{x}(t)|^2$$

$$\rightarrow \text{ENERGIA: } E = \int_{-\infty}^{+\infty} |\dot{x}(t)|^2 dt$$

$$\rightarrow \text{POTENZA MEDIA: } P \leq |\dot{x}(t)|^2 \geq \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |\dot{x}(t)|^2 dt$$